

Carnegie Mellon University

Course: 36-708

Statistical Methods for Machine Learning

Instructor: Larry Wasserman

This course is an advanced course focusing on the intersection of Statistics and Machine Learning. The goal is to study modern methods and the underlying theory for those methods. There are two pre-requisites for this course:

- 36-705 (Intermediate Statistical Theory)
- 36-707 (Regression)

36-708 Statistical Methods in Machine Learning

Syllabus, Spring 2019

<http://www.stat.cmu.edu/~larry/=sml>

Lectures: Tuesday and Thursday 1:30 - 2:50 pm (POS 152)

This course is an introduction to Statistical Machine Learning. The goal is to study modern methods and the underlying theory for those methods. There are two pre-requisites for this course:

1. 36-705 (Intermediate Statistical Theory)
2. 10-707 (Regression)

Contact Information

Instructor:

Larry Wasserman BH 132G 412-268-8727 larry@cmu.edu

Teaching Assistants:

The names and office hours for the TA's will be on the course website.

Office Hours

Larry Wasserman Tuesdays 12:00-1:00 pm Baker Hall 132G

Text

There is no text but course notes will be posted. Useful reference are:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman (2001). *The Elements of Statistical Learning*, Available at <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>.
2. Chris Bishop (2006). *Pattern Recognition and Machine Learning*.
3. Luc Devroye, László Györfi, Gábor Lugosi. (1996). *A probabilistic theory of pattern recognition*.
4. Györfi, Kohler, Krzyzak and Walk (2002). *A Distribution-Free Theory of Nonparametric Regression*.
5. Larry Wasserman (2004). *All of Statistics: A Concise Course in Statistical Inference*.
6. Larry Wasserman (2005). *All of Nonparametric Statistics*.

Grading

1. There will be four assignments. They are due **Fridays at 3:00 p.m.**. Hand them by uploading a pdf file to Canvas.
2. **Midterm Exam.** The date is **Thursday MARCH 7**.
3. **Project.** There will be a final project, described later in the syllabus.

Grading will be as follows:

50% Assignments

25% Midterm

25% Project

Policy on Collaboration

Collaboration on homework assignments with fellow students is encouraged. However, such collaboration should be clearly acknowledged, by listing the names of the students with whom you have had discussions concerning your solution. You may not, however, share written work or code after discussing a problem with others. The solutions should be written by you.

Topics

1. Introduction and Review
 - (a) Statistics versus ML
 - (b) concentration
 - (c) bias and variance
 - (d) minimax
 - (e) linear regression
 - (f) linear classification
 - (g) logistic regression
2. Nonparametric Inference
 - (a) Density Estimation
 - (b) Nonparametric Regression Regression
 - i. kernels
 - ii. local polynomial
 - iii. NN
 - iv. RKHS
 - (c) Nonparametric Classification
 - i. plug-in
 - ii. nn
 - iii. density-based
 - iv. kernelized SVM
 - v. trees
 - vi. random forests
3. High Dimensional Methods
 - (a) Forward stepwise regression
 - (b) Lasso
 - (c) Ridge Regression
 - (d) High dimensional classification
4. Clustering
5. Graphical models
6. Minimax theory
7. Causality
8. Dimension reduction: PCA, nonlinear
9. Other Possible Topics
 - (a) Mixture models
 - (b) Wasserstein distance and optimal transport
 - (c) boosting
 - (d) active learning

- (e) nonparametric bayes
- (f) deep learning
- (g) differential privacy
- (h) interactive data analysis
- (i) multinomials
- (j) statistics computational tradeoff
- (k) random matrices
- (l) conformal methods
- (m) interpolation

Course Calendar

The course calendar is posted on the course website and will be updated throughout the semester.

Project

The project involves picking a topic of interest, reading the relevant results in the area and then writing a short paper (8 pages) summarizing the key ideas in the area. You may focus on a single paper if you prefer. Your are NOT required to do new research.

The paper should include background, statement of important results, and brief proof outlines for the results. If appropriate, you should also include numerical experiments or an application with real data.

1. You may work by yourself or in teams of two.
2. The goals are (i) to summarize key results in literature on a particular topic **and** (ii) present a summary of the theoretical analysis (results and proof sketch) of the methods (iii) implement some of the main methods. You may develop new theory if you like but it is not required.
3. You will provide: (i) a proposal, (ii) a progress report and (iii) a final report.
4. The reports should be well-written.

Proposal. A one page proposal is due **February 8**. It should contain the following information: (1) project title, (2) team members, (3) precise description of the problem you are studying, (4) anticipated scope of the project, and (5) reading list. (Papers you will need to read).

Progress Report. Due **April 5**. Three pages. Include: (i) a high quality introduction, (ii) what have you done so far, (iii) what remains to be done and (iv) a clear description of the division of work among teammates, if applicable.

Final Report: Due **May 3**. The paper should be in NIPS format. (pdf only). **Maximum 8 pages.** No appendix is allowed. You should submit a pdf file electronically. It should have the following format:

1. Introduction. Motivation and a quick summary of the area.
2. Notation and Assumptions.
3. Key Results.
4. Proof outlines for the results.
5. Implementation (simulations or real data example.)
6. Conclusion. This includes comments on the meaning of the results and open questions.

36-708 Introduction and Review

1 Statistics versus ML

Statistics and ML are overlapping fields. Both address the same question: how do we extract information from data? But there are differences in emphasis. In particular, some topics get greater emphasis than others. Here are some examples:

More emphasis in ML	More emphasis in Stat	Common Areas
Bandits	Confidence Sets	Prediction (Regression and Classification)
Reinforcement Learning	Large Sample Theory	Probability Bounds (Concentration)
Efficient Computation	Statistical Optimality	Clustering
Deep Learning	Causality	Graphical Models

However, the lines between the two fields are blurry and will become increasingly so.

Another difference between the two fields is that ML researchers tend to publish short papers in conferences while Statisticians tend to publish long papers in journals. Each has advantages and disadvantages.

2 Concentration

Hoeffding's inequality:

Theorem 1 (Hoeffding) If Z_1, Z_2, \dots, Z_n are iid with mean μ and $\mathbb{P}(a \leq Z_i \leq b) = 1$, then for any $\epsilon > 0$

$$\mathbb{P}(|\bar{Z}_n - \mu| > \epsilon) \leq 2e^{-2n\epsilon^2/(b-a)^2} \quad (1)$$

where and $\bar{Z}_n = \frac{1}{n} \sum_{i=1}^n Z_i$.

VC Dimension. Let \mathcal{A} be a class of sets. If F is a finite set, let $s(\mathcal{A}, F)$ be the number of subset of F ‘picked out’ by \mathcal{A} . Define the growth function

$$s_n(\mathcal{A}) = \sup_{|F|=n} s(\mathcal{A}, F).$$

Note that $s_n(\mathcal{A}) \leq 2^n$. The *VC dimension* of a class of set \mathcal{A} is

$$\text{VC}(\mathcal{A}) = \sup \left\{ n : s_n(\mathcal{A}) = 2^n \right\}. \quad (2)$$

If the VC dimension is finite, then there is a phase transition in the growth function from exponential to polynomial:

Theorem 2 (Sauer's Theorem) Suppose that \mathcal{A} has finite VC dimension d . Then, for all $n \geq d$,

$$s(\mathcal{A}, n) \leq \left(\frac{en}{d}\right)^d. \quad (3)$$

Given data $Z_1, \dots, Z_n \sim P$. The empirical measure P_n is

$$P_n(A) = \frac{1}{n} \sum_i I(Z_i \in A).$$

Theorem 3 (Vapnik and Chervonenkis) Let \mathcal{A} be a class of sets. For any $t > \sqrt{2/n}$,

$$\mathbb{P} \left(\sup_{A \in \mathcal{A}} |P_n(A) - P(A)| > t \right) \leq 4 s(\mathcal{A}, 2n) e^{-nt^2/8} \quad (4)$$

and hence, with probability at least $1 - \delta$,

$$\sup_{A \in \mathcal{A}} |P_n(A) - P(A)| \leq \sqrt{\frac{8}{n} \log \left(\frac{4 s(\mathcal{A}, 2n)}{\delta} \right)}. \quad (5)$$

Hence, if \mathcal{A} has finite VC dimension d then

$$\sup_{A \in \mathcal{A}} |P_n(A) - P(A)| \leq \sqrt{\frac{8}{n} \left(\log \left(\frac{4}{\delta} \right) + d \log \left(\frac{ne}{d} \right) \right)}. \quad (6)$$

Bernstein's inequality is a more refined inequality than Hoeffding's inequality. It is especially useful when the variance of Y is small. Suppose that Y_1, \dots, Y_n are iid with mean μ , $\text{Var}(Y_i) \leq \sigma^2$ and $|Y_i| \leq M$. Then

$$\mathbb{P}(|\bar{Y} - \mu| > \epsilon) \leq 2 \exp \left\{ -\frac{n\epsilon^2}{2\sigma^2 + 2M\epsilon/3} \right\}. \quad (7)$$

It follows that

$$P \left(|\bar{Y} - \mu| > \frac{t}{n\epsilon} + \frac{\epsilon\sigma^2}{2(1-c)} \right) \leq e^{-t}$$

for small enough ϵ and c .

3 Probability

1. $X_n \xrightarrow{P} 0$ means that means that, for every $\epsilon > 0$ $\mathbb{P}(|X_n| > \epsilon) \rightarrow 0$ as $n \rightarrow \infty$.

2. $X_n \rightsquigarrow Z$ means that $\mathbb{P}(X_n \leq z) \rightarrow \mathbb{P}(Z \leq z)$ at all continuity points z .
3. $X_n = O_P(a_n)$ means that, X_n/a_n is bounded in probability: for every $\epsilon > 0$ there is an $M > 0$ such that, for all large n , $\mathbb{P}\left(\left|\frac{X_n}{a_n}\right| > M\right) \leq \epsilon$.
4. $X_n = o_p(a_n)$ means that X_n/a_n goes to 0 in probability: for every $\epsilon > 0$

$$\mathbb{P}\left(\left|\frac{X_n}{a_n}\right| > \epsilon\right) \rightarrow 0 \quad \text{as } n \rightarrow \infty.$$

5. Law of large numbers: $X_1, \dots, X_n \sim P$ then

$$\overline{X}_n \xrightarrow{P} \mu$$

where $\overline{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X_i]$.

6. Central limit theorem: $X_1, \dots, X_n \sim P$ then

$$\frac{\sqrt{n}(\overline{X}_n - \mu)}{\sigma} \rightsquigarrow N(0, 1)$$

where $\sigma^2 = \text{Var}(X_i)$.

4 Basic Statistics

1. **Bias and Variance.** Let $\hat{\theta}$ be an estimator of θ . Then

$$\mathbb{E}(\hat{\theta} - \theta)^2 = \text{bias}^2 + \text{Var}$$

where $\text{bias} = \mathbb{E}[\hat{\theta}] - \theta$ and $\text{Var} = \text{Var}(\hat{\theta})$. In many cases there is a **bias-variance trade-off**. In parametric problems, we typically have that the standard deviation is $O(n^{-1/2})$ but the bias is $O(1/n)$ so the variability dominates. In nonparametric problems this is no longer true. We have to choose tuning parameters in classifiers and estimators to balance the bias and variance.

2. A set of distributions \mathcal{P} is a **statistical model**. They can be small (parametric models) or large (nonparametric models).
3. **Confidence Sets.** Let $X_1, \dots, X_n \sim P$ where $P \in \mathcal{P}$. Let $\theta = T(P)$ be some quantity of interest, Then $C_n = C(X_1, \dots, X_n)$ is a $1 - \alpha$ confidence set if

$$\inf_{P \in \mathcal{P}} P(T(P) \in C_n) \geq 1 - \alpha.$$

4. **Maximum Likelihood.** Parametric model $\{p_\theta : \theta \in \Theta\}$. We also write $p_\theta(x) = p(x; \theta)$. Let $X_1, \dots, X_n \sim p_\theta$. MLE $\hat{\theta}_n$ (maximum likelihood estimator) maximizes the likelihood function

$$\mathcal{L}(\theta) = \prod_{i=1}^n p(X_i; \theta).$$

5. Fisher information $I_n(\theta) = nI(\theta)$ where

$$I(\theta) = -\mathbb{E} \left[\frac{\partial^2 \log p(X; \theta)}{\partial \theta^2} \right].$$

6. Then

$$\frac{\hat{\theta}_n - \theta}{s_n} \rightsquigarrow N(0, 1)$$

where $s_n = \sqrt{\frac{1}{nI(\hat{\theta})}}$.

7. Asymptotic $1 - \alpha$ confidence interval $C_n = \hat{\theta}_n \pm z_{\alpha/2} s_n$. Then

$$\mathbb{P}(\theta \in C_n) \rightarrow 1 - \alpha.$$

5 Minimaxity

Let \mathcal{P} be a set of distributions. Let θ be a parameter and let $L(\hat{\theta}, \theta)$ be a loss function. The **minimax risk** is

$$R_n = \inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[L(\hat{\theta}, \theta)].$$

If $\sup_{P \in \mathcal{P}} \mathbb{E}_P[L(\hat{\theta}, \theta)] = R_n$ then $\hat{\theta}$ is a minimax estimator.

For example, if $X_1, \dots, X_n \sim N(\theta, 1)$ and $L(\hat{\theta}, \theta) = (\hat{\theta} - \theta)^2$ then the minimax risk is $1/n$ and the minimax estimator is \bar{X}_n .

As another example, if $X_1, \dots, X_n \sim p$ where $X_i \in \mathbb{R}^d$, $L(\hat{p}, p) = \int (\hat{p} - p)^2$ and $p \in \mathcal{P}$, the set of densities with bounded second derivatives, then $R_n = (C/n)^{4/(4+d)}$. The kernel density estimator is minimax.

6 Regression

1. $Y \in \mathbb{R}$, $X \in \mathbb{R}^d$ and prediction risk is

$$\mathbb{E}(Y - m(X))^2.$$

We write $X = (X(1), \dots, X(d))$.

2. Minimizer is $m(x) = \mathbb{E}(Y|X = x)$.

3. Best linear predictor: minimize

$$\mathbb{E}(Y - \beta^T X)^2$$

where $X(1) = 1$ so that β_1 is the intercept. Minimizer is

$$\beta = \Lambda^{-1} \alpha$$

where $\Lambda(j, k) = \mathbb{E}[X(j)X(k)]$ and $\alpha(j) = \mathbb{E}(YX(j))$.

4. The data are

$$(X_1, Y_1), \dots, (X_n, Y_n).$$

Given new X predict Y .

5. Minimize training error

$$\hat{R}(\beta) = \frac{1}{n} \sum_i (Y_i - \beta^T X_i)^2.$$

Solution: least squares:

$$\hat{\beta} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T Y$$

where $\mathbb{X}(i, j) = X_i(j)$.

6. Fitted values $\hat{Y} = \mathbb{X}\hat{\beta} = HY$ where $H = \mathbb{X}(\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T$ is the hat matrix: the projector onto the column space of \mathbb{X} .

7. Bias-Variance tradeoff: Write $Y = m(X) + \epsilon$ and let $\hat{Y} = \hat{m}(X)$ where $\hat{m}(x) = x^T \hat{\beta}$. Then

$$R = \mathbb{E}(\hat{Y} - Y)^2 = \sigma^2 + \int b^2(x)p(x)dx + \int v(x)p(x)dx$$

where $b(x) = \mathbb{E}[\hat{m}(x)] - m(x)$, $v(x) = \text{Var}(\hat{m}(x))$ and $\sigma^2 = \text{Var}(\epsilon)$.

7 Classification

1. $X \in \mathbb{R}^d$ and $Y \in \{0, 1\}$.
2. Classifier $h : \mathbb{R}^d \rightarrow \{0, 1\}$.
3. Prediction risk:

$$R(h) = \mathbb{P}(Y \neq h(X)).$$

The **Bayes rule** minimizes $R(h)$:

$$h(x) = I(m(x) > 1/2) = I(\pi_1 p_1(x) > \pi_0 p_0(x))$$

where $m(x) = \mathbb{P}(Y = 1|X = x)$, $\pi_1 = \mathbb{P}(Y = 1)$, $\pi_0 = \mathbb{P}(Y = 0)$, $p_1(x) = p(x|Y = 1)$ and $p_0(x) = p(x|Y = 0)$.

4. **Re-coded loss.** If we code Y as $Y \in \{-1, +1\}$. then many classifiers can be written as

$$h(x) = \text{sign}(\psi(x))$$

for some ψ . For linear classifiers, $\psi(x) = \beta^T x$. Then the loss can be written as $I(Y \neq h(X)) = I(Y\psi(X) < 0)$ and risk is

$$R = \mathbb{P}(Y \neq h(X)) = \mathbb{P}(Y\psi(X) < 0)$$

5. **Linear Classifiers.** A linear classifier has the form $h_\beta(x) = I(\beta^T x > 0)$. (I am including a intercept in x . In other words $x = (1, x(2), \dots, x(d))$.) Given data $(X_1, Y_1), \dots, (X_n, Y_n)$ there are several ways to estimate a linear classifier:

(a) Empirical risk minimization (ERM): Choose $\hat{\beta}$ to minimize

$$R_n(\beta) = \frac{1}{n} \sum_{i=1}^n I(Y_i \neq h_\beta(X_i)).$$

(b) Logistic regression: use the model

$$P(Y = 1 | X = x) = \frac{e^{\beta^T x}}{1 + e^{\beta^T x}} \equiv p(x, \beta).$$

So $Y_i \sim \text{Benoulli}(p(X_i, \beta))$. The likelihood function is

$$L(\beta) = \prod_i p(X_i, \beta)^{Y_i} (1 - p(X_i, \beta))^{1-Y_i}.$$

The log-likelihood is strictly concave. So we have find the maximizer $\hat{\beta}$ easily. It is easy to check that the classifier $I(p_{x, \hat{\beta}} > 1/2)$ is linear.

(c) SVM (support vector machine). Code Y as $+1$ or -1 . We can write the classifier as $h_\beta(x) = \text{sign}(\psi_\beta(x))$ where $\psi_\beta(x) = x^T \beta$. As we said above, the loss can be written as $I(Y \neq h(X)) = I(Y\psi(X) < 0)$. Now replace the nonconvex loss $I(Y\psi(X) < 0)$ with the hinge-loss $[1 - Y_i\psi_\beta(X_i)]_+$. We minimize the regularized loss

$$\sum_{i=1}^n [1 - Y_i\psi_\beta(X_i)]_+ + \lambda \|\beta\|^2.$$

6. The SVM is an example of the general idea of replacing the true loss with a surrogate loss that is easier to minimize. Replacing $I(Y\psi(X) < 0)$ with

$$L(Y, \psi(X)) = \log(1 + \exp(-Y\psi(X)))$$

gives back logistic regression. The adaboost algorithm uses

$$L(Y, \psi(X)) = \exp(-Y\psi(X)).$$

And, as we said above, the SVM uses the hinge loss

$$L(Y, \psi(X)) = [1 - Y\psi(X)]_+.$$

Density Estimation

36-708

1 Introduction

Let X_1, \dots, X_n be a sample from a distribution P with density p . The goal of nonparametric density estimation is to estimate p with as few assumptions about p as possible. We denote the estimator by \hat{p} . The estimator will depend on a smoothing parameter h and choosing h carefully is crucial. To emphasize the dependence on h we sometimes write \hat{p}_h .

Density estimation used for: regression, classification, clustering and unsupervised prediction. For example, if $\hat{p}(x, y)$ is an estimate of $p(x, y)$ then we get the following estimate of the regression function:

$$\hat{m}(x) = \int y\hat{p}(y|x)dy$$

where $\hat{p}(y|x) = \hat{p}(y, x)/\hat{p}(x)$. For classification, recall that the Bayes rule is

$$h(x) = I(p_1(x)\pi_1 > p_0(x)\pi_0)$$

where $\pi_1 = \mathbb{P}(Y = 1)$, $\pi_0 = \mathbb{P}(Y = 0)$, $p_1(x) = p(x|y = 1)$ and $p_0(x) = p(x|y = 0)$. Inserting sample estimates of π_1 and π_0 , and density estimates for p_1 and p_0 yields an estimate of the Bayes rule. For clustering, we look for the high density regions, based on an estimate of the density. Many classifiers that you are familiar with can be re-expressed this way. Unsupervised prediction is discussed in Section 9. In this case we want to predict X_{n+1} from X_1, \dots, X_n .

Example 1 (Bart Simpson) *The top left plot in Figure 1 shows the density*

$$p(x) = \frac{1}{2}\phi(x; 0, 1) + \frac{1}{10} \sum_{j=0}^4 \phi(x; (j/2) - 1, 1/10) \quad (1)$$

where $\phi(x; \mu, \sigma)$ denotes a Normal density with mean μ and standard deviation σ . Marron and Wand (1992) call this density “the claw” although we will call it the Bart Simpson density. Based on 1,000 draws from p , we computed a kernel density estimator, described later. The estimator depends on a tuning parameter called the bandwidth. The top right plot is based on a small bandwidth h which leads to undersmoothing. The bottom right plot is based on a large bandwidth h which leads to oversmoothing. The bottom left plot is based on a bandwidth h which was chosen to minimize estimated risk. This leads to a much more reasonable density estimate.

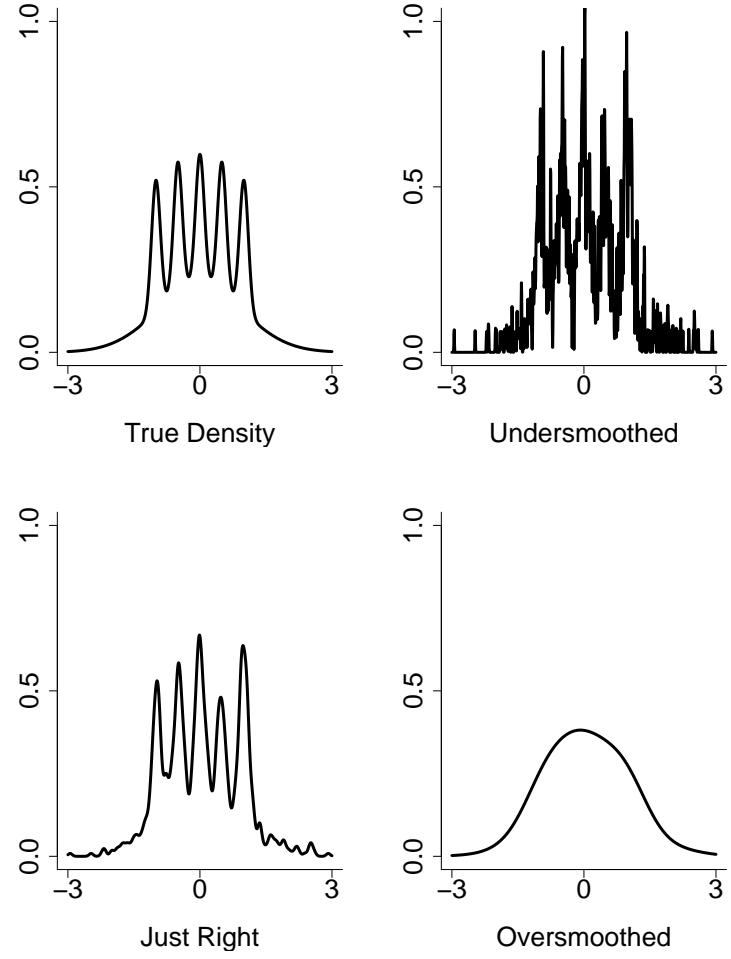


Figure 1: The Bart Simpson density from Example 1. Top left: true density. The other plots are kernel estimators based on $n = 1,000$ draws. Bottom left: bandwidth $h = 0.05$ chosen by leave-one-out cross-validation. Top right: bandwidth $h/10$. Bottom right: bandwidth $10h$.

2 Loss Functions

The most commonly used loss function is the L_2 loss

$$\int (\hat{p}(x) - p(x))^2 dx = \int \hat{p}^2(x) dx - 2 \int \hat{p}(x)p(x) + \int p^2(x) dx.$$

The risk is $R(p, \hat{p}) = \mathbb{E}(L(p, \hat{p}))$.

Devroye and Györfi (1985) make a strong case for using the L_1 norm

$$\|\hat{p} - p\|_1 \equiv \int |\hat{p}(x) - p(x)| dx$$

as the loss instead of L_2 . The L_1 loss has the following nice interpretation. If P and Q are distributions define the total variation metric

$$d_{TV}(P, Q) = \sup_A |P(A) - Q(A)|$$

where the supremum is over all measurable sets. Now if P and Q have densities p and q then

$$d_{TV}(P, Q) = \frac{1}{2} \int |p - q| = \frac{1}{2} \|p - q\|_1. \quad \mathbf{H}$$

Thus, if $\int |p - q| < \delta$ then we know that $|P(A) - Q(A)| < \delta/2$ for all A . Also, the L_1 norm is transformation invariant. Suppose that T is a one-to-one smooth function. Let $Y = T(X)$. Let p and q be densities for X and let \tilde{p} and \tilde{q} be the corresponding densities for Y . Then

$$\int |p(x) - q(x)| dx = \int |\tilde{p}(y) - \tilde{q}(y)| dy. \quad \mathbf{H}$$

Hence the distance is unaffected by transformations. The L_1 loss is, in some sense, a much better loss function than L_2 for density estimation. But it is much more difficult to deal with. For now, we will focus on L_2 loss. But we may discuss L_1 loss later.

Another loss function is the Kullback-Leibler loss $\int p(x) \log p(x)/q(x) dx$. This is not a good loss function to use for nonparametric density estimation. The reason is that the Kullback-Leibler loss is completely dominated by the tails of the densities. \mathbf{H}

3 Histograms

Perhaps the simplest density estimators are histograms. For convenience, assume that the data X_1, \dots, X_n are contained in the unit cube $\mathcal{X} = [0, 1]^d$ (although this assumption is not crucial). Divide \mathcal{X} into bins, or sub-cubes, of size h . **We discuss methods for choosing**

h later. There are $N \approx (1/h)^d$ such bins and each has volume h^d . Denote the bins by B_1, \dots, B_N . The histogram density estimator is

$$\hat{p}_h(x) = \sum_{j=1}^N \frac{\hat{\theta}_j}{h^d} I(x \in B_j) \quad (2)$$

where

$$\hat{\theta}_j = \frac{1}{n} \sum_{i=1}^n I(X_i \in B_j)$$

is the fraction of data points in bin B_j . Now we bound the bias and variance of \hat{p}_h . We will assume that $p \in \mathcal{P}(L)$ where

$$\mathcal{P}(L) = \left\{ p : |p(x) - p(y)| \leq L \|x - y\|, \text{ for all } x, y \right\}. \quad (3)$$

First we bound the bias. Let $\theta_j = P(X \in B_j) = \int_{B_j} p(u) du$. For any $x \in B_j$,

$$p_h(x) \equiv \mathbb{E}(\hat{p}_h(x)) = \frac{\theta_j}{h^d} \quad (4)$$

and hence

$$p(x) - p_h(x) = p(x) - \frac{\int_{B_j} p(u) du}{h^d} = \frac{1}{h^d} \int (p(x) - p(u)) du.$$

Thus,

$$|p(x) - p_h(x)| \leq \frac{1}{h^d} \int |p(x) - p(u)| du \leq \frac{1}{h^d} L h \sqrt{d} \int du = L h \sqrt{d}$$

where we used the fact that if $x, u \in B_j$ then $\|x - u\| \leq \sqrt{d}h$.

Now we bound the variance. Since p is Lipschitz on a compact set, it is bounded. Hence, $\theta_j = \int_{B_j} p(u) du \leq C \int_{B_j} du = Ch^d$ for some C . Thus, the variance is

$$\text{Var}(\hat{p}_h(x)) = \frac{1}{h^{2d}} \text{Var}(\hat{\theta}_j) = \frac{\theta_j(1 - \theta_j)}{nh^{2d}} \leq \frac{\theta_j}{nh^{2d}} \leq \frac{C}{nh^d}.$$

We conclude that the L_2 risk is bounded by

$$\sup_{p \in \mathcal{P}(L)} R(p, \hat{p}) = \int (\mathbb{E}(\hat{p}_h(x) - p(x))^2 \leq L^2 h^2 d + \frac{C}{nh^d}. \quad (5)$$

The upper bound is minimized by choosing $h = (\frac{C}{L^2 nd})^{\frac{1}{d+2}}$. (Later, we shall see a more practical way to choose h .) With this choice,

$$\sup_{p \in \mathcal{P}(L)} R(p, \hat{p}) \leq C_0 \left(\frac{1}{n} \right)^{\frac{2}{d+2}}$$

where $C_0 = L^2d(C/(L^2d))^{2/(d+2)}$.

Later, we will prove the following theorem which shows that this upper bound is tight. Specifically:

Theorem 2 *There exists a constant $C > 0$ such that*

$$\inf_{\hat{p}} \sup_{P \in \mathcal{P}(L)} \mathbb{E} \int (\hat{p}(x) - p(x))^2 dx \geq C \left(\frac{1}{n} \right)^{\frac{2}{d+2}}. \quad (6)$$

3.1 Concentration Analysis For Histograms

Let us now derive a concentration result for \hat{p}_h . We will bound

$$\sup_{P \in \mathcal{P}} P^n(\|\hat{p}_h - p\|_\infty > \epsilon)$$

where $\|f\|_\infty = \sup_x |f(x)|$. Assume that $\epsilon \leq 1$. First, note that

$$\mathbb{P}(\|\hat{p}_h - p_h\|_\infty > \epsilon) = \mathbb{P}\left(\max_j \left| \frac{\hat{\theta}_j}{h^d} - \frac{\theta_j}{h^d} \right| > \epsilon\right) = \mathbb{P}(\max_j |\hat{\theta}_j - \theta_j| > h^d \epsilon) \leq \sum_j \mathbb{P}(|\hat{\theta}_j - \theta_j| > h^d \epsilon).$$

Using Bernstein's inequality and the fact that $\theta_j(1 - \theta_j) \leq \theta_j \leq Ch^d$,

$$\begin{aligned} \mathbb{P}(|\hat{\theta}_j - \theta_j| > h^d \epsilon) &\leq 2 \exp\left(-\frac{1}{2} \frac{n\epsilon^2 h^{2d}}{\theta_j(1 - \theta_j) + \epsilon h^d / 3}\right) \\ &\leq 2 \exp\left(-\frac{1}{2} \frac{n\epsilon^2 h^{2d}}{Ch^d + \epsilon h^d / 3}\right) \\ &\leq 2 \exp(-c n \epsilon^2 h^d) \end{aligned}$$

where $c = 1/(2(C + 1/3))$. By the union bound and the fact that $N \leq (1/h)^d$,

$$\mathbb{P}(|\hat{\theta}_j - \theta_j| > h^d \epsilon) \leq 2h^{-d} \exp(-c n \epsilon^2 h^d) \equiv \pi_n.$$

Earlier we saw that $\sup_x |p(x) - p_h(x)| \leq L\sqrt{dh}$. Hence, with probability at least $1 - \pi_n$,

$$\|\hat{p}_h - p\|_\infty \leq \|\hat{p}_h - p_h\|_\infty + \|p_h - p\|_\infty \leq \epsilon + L\sqrt{dh}. \quad (7)$$

Now set

$$\epsilon = \sqrt{\frac{1}{cnh^d} \log\left(\frac{2}{\delta h^d}\right)}.$$

Then, with probability at least $1 - \delta$,

$$\|\hat{p}_h - p\|_\infty \leq \sqrt{\frac{1}{cnh^d} \log \left(\frac{2}{\delta h^d} \right)} + L\sqrt{dh}. \quad (8)$$

Choosing $h = (c_2/n)^{1/(2+d)}$ we conclude that, with probability at least $1 - \delta$,

$$\|\hat{p}_h - p\|_\infty \leq \sqrt{c^{-1} n^{-\frac{2}{2+d}} \left[\log \left(\frac{2}{\delta} \right) + \left(\frac{2}{2+d} \right) \log n \right]} + L\sqrt{dn^{-\frac{1}{2+d}}} = O \left(\left(\frac{\log n}{n} \right)^{\frac{1}{2+d}} \right). \quad (9)$$

4 Kernel Density Estimation

A one-dimensional smoothing kernel is any smooth function K such that $\int K(x) dx = 1$, $\int xK(x)dx = 0$ and $\sigma_K^2 \equiv \int x^2 K(x)dx > 0$. *Smoothing kernels* should not be confused with *Mercer kernels* which we discuss later. Some commonly used kernels are the following:

Boxcar: $K(x) = \frac{1}{2}I(x)$	Gaussian: $K(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$
Epanechnikov: $K(x) = \frac{3}{4}(1-x^2)I(x)$	Tricube: $K(x) = \frac{70}{81}(1- x ^3)^3I(x)$

where $I(x) = 1$ if $|x| \leq 1$ and $I(x) = 0$ otherwise. These kernels are plotted in Figure 2. Two commonly used multivariate kernels are $\prod_{j=1}^d K(x_j)$ and $K(\|x\|)$.

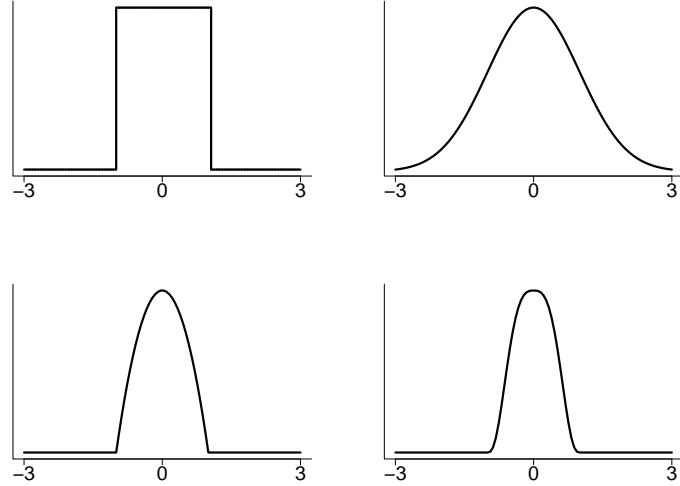


Figure 2: Examples of smoothing kernels: boxcar (top left), Gaussian (top right), Epanechnikov (bottom left), and tricube (bottom right).

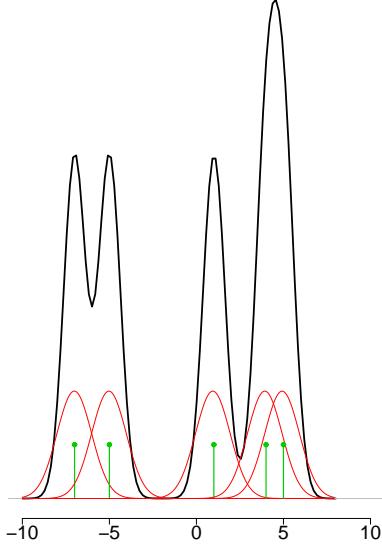


Figure 3: A kernel density estimator \hat{p} . At each point x , $\hat{p}(x)$ is the average of the kernels centered over the data points X_i . The data points are indicated by short vertical bars. The kernels are not drawn to scale.

Suppose that $X \in \mathbb{R}^d$. Given a kernel K and a positive number h , called the bandwidth, the kernel density estimator is defined to be

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^d} K\left(\frac{\|x - X_i\|}{h}\right). \quad (10)$$

More generally, we define

$$\hat{p}_H(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - X_i)$$

where H is a positive definite bandwidth matrix and $K_H(x) = |H|^{-1/2}K(H^{-1/2}x)$. For simplicity, we will take $H = h^2I$ and we get back the previous formula.

Sometimes we write the estimator as \hat{p}_h to emphasize the dependence on h . In the multivariate case the coordinates of X_i should be standardized so that each has the same variance, since the norm $\|x - X_i\|$ treats all coordinates as if they are on the same scale.

The kernel estimator places a smoothed out lump of mass of size $1/n$ over each data point X_i ; see Figure 3. The choice of kernel K is not crucial, but the choice of bandwidth h is important. Small bandwidths give very rough estimates while larger bandwidths give smoother estimates.

4.1 Risk Analysis

In this section we examine the accuracy of kernel density estimation. We will first need a few definitions.

Assume that $X_i \in \mathcal{X} \subset \mathbb{R}^d$ where \mathcal{X} is compact. Let β and L be positive numbers. Given a vector $s = (s_1, \dots, s_d)$, define $|s| = s_1 + \dots + s_d$, $s! = s_1! \cdots s_d!$, $x^s = x_1^{s_1} \cdots x_d^{s_d}$ and

$$D^s = \frac{\partial^{s_1+\dots+s_d}}{\partial x_1^{s_1} \cdots \partial x_d^{s_d}}.$$

Let β be a positive integer. Define the Hölder class

$$\Sigma(\beta, L) = \left\{ g : |D^s g(x) - D^s g(y)| \leq L \|x - y\|, \text{ for all } s \text{ such that } |s| = \beta - 1, \text{ and all } x, y \right\}. \quad (11)$$

For example, if $d = 1$ and $\beta = 2$ this means that

$$|g'(x) - g'(y)| \leq L |x - y|, \quad \text{for all } x, y.$$

The most common case is $\beta = 2$; roughly speaking, this means that the functions have bounded second derivatives.

If $g \in \Sigma(\beta, L)$ then $g(x)$ is close to its Taylor series approximation:

$$|g(u) - g_{x,\beta}(u)| \leq L \|u - x\|^\beta \quad (12)$$

where

$$g_{x,\beta}(u) = \sum_{|s| \leq \beta} \frac{(u - x)^s}{s!} D^s g(x). \quad (13)$$

In the common case of $\beta = 2$, this means that

$$\left| p(u) - [p(x) + (x - u)^T \nabla p(x)] \right| \leq L \|x - u\|^2.$$

Assume now that the kernel K has the form $K(x) = G(x_1) \cdots G(x_d)$ where G has support on $[-1, 1]$, $\int G = 1$, $\int |G|^p < \infty$ for any $p \geq 1$, $\int |t|^\beta |K(t)| dt < \infty$ and $\int t^s K(t) dt = 0$ for $s \leq \beta$.

An example of a kernel that satisfies these conditions for $\beta = 2$ is $G(x) = (3/4)(1 - x^2)$ for $|x| \leq 1$. Constructing a kernel that satisfies $\int t^s K(t) dt = 0$ for $\beta > 2$ requires using kernels that can take negative values.

Let $p_h(x) = \mathbb{E}[\hat{p}_h(x)]$. The next lemma provides a bound on the bias $p_h(x) - p(x)$.

Lemma 3 *The bias of \hat{p}_h satisfies:*

$$\sup_{p \in \Sigma(\beta, L)} |p_h(x) - p(x)| \leq ch^\beta \quad (14)$$

for some c .

Proof. We have

$$\begin{aligned} |p_h(x) - p(x)| &= \int \frac{1}{h^d} K(\|u - x\|/h) p(u) du - p(x) \\ &= \left| \int K(\|v\|) (p(x + hv) - p(x)) dv \right| \\ &\leq \left| \int K(\|v\|) (p(x + hv) - p_{x,\beta}(x + hv)) dv \right| + \left| \int K(\|v\|) (p_{x,\beta}(x + hv) - p(x)) dv \right|. \end{aligned}$$

The first term is bounded by $Lh^\beta \int K(s)|s|^\beta$ since $p \in \Sigma(\beta, L)$. The second term is 0 from the properties on K since $p_{x,\beta}(x + hv) - p(x)$ is a polynomial of degree β (with no constant term). \square

Next we bound the variance.

Lemma 4 *The variance of \hat{p}_h satisfies:*

$$\sup_{p \in \Sigma(\beta, L)} \text{Var}(\hat{p}_h(x)) \leq \frac{c}{nh^d} \quad (15)$$

for some $c > 0$.

Proof. We can write $\hat{p}(x) = n^{-1} \sum_{i=1}^n Z_i$ where $Z_i = \frac{1}{h^d} K\left(\frac{\|x-X_i\|}{h}\right)$. Then,

$$\begin{aligned} \text{Var}(Z_i) &\leq \mathbb{E}(Z_i^2) = \frac{1}{h^{2d}} \int K^2\left(\frac{\|x-u\|}{h}\right) p(u) du = \frac{h^d}{h^{2d}} \int K^2(\|v\|) p(x+hv) dv \\ &\leq \frac{\sup_x p(x)}{h^d} \int K^2(\|v\|) dv \leq \frac{c}{h^d} \end{aligned}$$

for some c since the densities in $\Sigma(\beta, L)$ are uniformly bounded. The result follows. \square

Since the mean squared error is equal to the variance plus the bias squared we have:

Theorem 5 *The L_2 risk is bounded above, uniformly over $\Sigma(\beta, L)$, by $h^{4\beta} + \frac{1}{nh^d}$ (up to constants). If $h \asymp n^{-1/(2\beta+d)}$ then*

$$\sup_{p \in \Sigma(\beta, L)} \mathbb{E} \int (\hat{p}_h(x) - p(x))^2 dx \preceq \left(\frac{1}{n}\right)^{\frac{2\beta}{2\beta+d}}. \quad (16)$$

When $\beta = 2$ and $h \asymp n^{-1/(4+d)}$ we get the rate $n^{-4/(4+d)}$.

4.2 Minimax Bound

According to the next theorem, there does not exist an estimator that converges faster than $O(n^{-2\beta/(2\beta+d)})$. We state the result for integrated L_2 loss although similar results hold for other loss functions and other function spaces. We will prove this later in the course.

Theorem 6 *There exists C depending only on β and L such that*

$$\inf_{\hat{p}} \sup_{p \in \Sigma(\beta, L)} \mathbb{E}_p \int (\hat{p}(x) - p(x))^2 dx \geq C \left(\frac{1}{n} \right)^{\frac{2\beta}{2\beta+d}}. \quad (17)$$

Theorem 6 together with (16) imply that kernel estimators are rate minimax.

4.3 Concentration Analysis of Kernel Density Estimator

Now we state a result which says how fast $\hat{p}(x)$ concentrates around $p(x)$. First, recall Bernstein's inequality: Suppose that Y_1, \dots, Y_n are iid with mean μ , $\text{Var}(Y_i) \leq \sigma^2$ and $|Y_i| \leq M$. Then

$$\mathbb{P}(|\bar{Y} - \mu| > \epsilon) \leq 2 \exp \left\{ -\frac{n\epsilon^2}{2\sigma^2 + 2M\epsilon/3} \right\}. \quad (18)$$

Theorem 7 *For all small $\epsilon > 0$,*

$$\mathbb{P}(|\hat{p}(x) - p_h(x)| > \epsilon) \leq 2 \exp \left\{ -cnh^d\epsilon^2 \right\}. \quad (19)$$

Hence, for any $\delta > 0$,

$$\sup_{p \in \Sigma(\beta, L)} \mathbb{P} \left(|\hat{p}(x) - p(x)| > \sqrt{\frac{C \log(2/\delta)}{nh^d}} + ch^\beta \right) < \delta \quad (20)$$

for some constants C and c . If $h \asymp n^{-1/(2\beta+d)}$ then

$$\sup_{p \in \Sigma(\beta, L)} \mathbb{P} \left(|\hat{p}(x) - p(x)|^2 > \frac{c}{n^{2\beta/(2\beta+d)}} \right) < \delta.$$

Note that the last statement follows from the bias-variance calculation followed by Markov's inequality. The first statement does not.

Proof. By the triangle inequality,

$$|\hat{p}(x) - p(x)| \leq |\hat{p}(x) - p_h(x)| + |p_h(x) - p(x)| \quad (21)$$

where $p_h(x) = \mathbb{E}(\hat{p}(x))$. From Lemma 3, $|p_h(x) - p(x)| \leq ch^\beta$ for some c . Now $\hat{p}(x) = n^{-1} \sum_{i=1}^n Z_i$ where

$$Z_i = \frac{1}{h^d} K\left(\frac{\|x - X_i\|}{h}\right).$$

Note that $|Z_i| \leq c_1/h^d$ where $c_1 = K(0)$. Also, $\text{Var}(Z_i) \leq c_2/h^d$ from Lemma 4. Hence, by Bernstein's inequality,

$$\mathbb{P}(|\hat{p}(x) - p_h(x)| > \epsilon) \leq 2 \exp\left\{-\frac{n\epsilon^2}{2c_2h^{-d} + 2c_1h^{-d}\epsilon/3}\right\} \leq 2 \exp\left\{-\frac{nh^d\epsilon^2}{4c_2}\right\}$$

whenever $\epsilon \leq 3c_2/c_1$. If we choose $\epsilon = \sqrt{C \log(2/\delta)/(nh^d)}$ where $C = 4c_2$ then

$$\mathbb{P}\left(|\hat{p}(x) - p_h(x)| > \sqrt{\frac{C}{nh^d}}\right) \leq \delta.$$

The result follows from (21). \square

4.4 Concentration in L_∞

Theorem 7 shows that, for each x , $\hat{p}(x)$ is close to $p(x)$ with high probability. We would like a version of this result that holds uniformly over all x . That is, we want a concentration result for

$$\|\hat{p} - p\|_\infty = \sup_x |\hat{p}(x) - p(x)|.$$

We can write

$$\|\hat{p}_h - p\|_\infty \leq \|\hat{p}_h - p_h\|_\infty + \|p_h - p\|_\infty \leq \|\hat{p}_h - p_h\|_\infty + ch^\beta.$$

We can bound the first term using something called *bracketing* together with Bernstein's theorem to prove that,

$$\mathbb{P}(\|\hat{p}_h - p_h\|_\infty > \epsilon) \leq 4 \left(\frac{C}{h^{d+1}\epsilon}\right)^d \exp\left(-\frac{3n\epsilon^2 h^d}{28K(0)}\right). \quad (22)$$

An alternative approach is to replace Bernstein's inequality with a more sophisticated inequality due to Talagrand. We follow the analysis in Giné and Guillou (2002). Let

$$\mathcal{F} = \left\{K\left(\frac{x - \cdot}{h}\right), x \in \mathbb{R}^d, h > 0\right\}.$$

We assume there exists positive numbers A and v such that

$$\sup_P N(\mathcal{F}_h, L_2(P), \epsilon \|F\|_{L_2(P)}) \leq \left(\frac{A}{\epsilon}\right)^v, \quad (23)$$

where $N(T, d, \epsilon)$ denotes the ϵ -covering number of the metric space (T, d) , F is the envelope function of \mathcal{F} and the supremum is taken over the set of all probability measures on \mathbb{R}^d . The quantities A and v are called the VC characteristics of \mathcal{F}_h .

Theorem 8 (Giné and Guillou 2002) *Assume that the kernel satisfies the above property.*

1. *Let $h > 0$ be fixed. Then, there exist constants $c_1 > 0$ and $c_2 > 0$ such that, for all small $\epsilon > 0$ and all large n ,*

$$\mathbb{P} \left\{ \sup_{x \in \mathbb{R}^d} |\hat{p}_h(x) - p_h(x)| > \epsilon \right\} \leq c_1 \exp \left\{ -c_2 n h^d \epsilon^2 \right\}. \quad (24)$$

2. *Let $h_n \rightarrow 0$ as $n \rightarrow \infty$ in such a way that $\frac{nh_n^d}{|\log h_n^d|} \rightarrow \infty$. Let*

$$\epsilon_n \geq \sqrt{\frac{|\log h_n|}{nh_n^d}}. \quad (25)$$

Then, for all n large enough, (24) holds with h and ϵ replaced by h_n and ϵ_n , respectively.

The above theorem imposes minimal assumptions on the kernel K and, more importantly, on the probability distribution P , whose density is not required to be bounded or smooth, and, in fact, may not even exist. Combining the above theorem with Lemma 3 we have the following result.

Theorem 9 *Suppose that $p \in \Sigma(\beta, L)$. Fix any $\delta > 0$. Then*

$$\mathbb{P} \left(\sup_x |\hat{p}(x) - p(x)| > \sqrt{\frac{C \log n}{nh^d}} + ch^\beta \right) < \delta$$

for some constants C and c where C depends on δ . Choosing $h \asymp \log n / n^{-1/(2\beta+d)}$ we have

$$\mathbb{P} \left(\sup_x |\hat{p}(x) - p(x)|^2 > \frac{C \log n}{n^{2\beta/(2\beta+d)}} \right) < \delta.$$

4.5 Boundary Bias

We have ignored what happens near the boundary of the sample space. If x is $O(h)$ close to the boundary, the bias is $O(h)$ instead of $O(h^2)$. There are a variety of fixes including: data reflection, transformations, boundary kernels, local likelihood.

4.6 Confidence Bands and the CLT

Consider first a single point x . Let $s_n(x) = \sqrt{\text{Var}(\hat{p}_h(x))}$. The CLT implies that

$$Z_n(x) \equiv \frac{\hat{p}_h(x) - p_h(x)}{s_n(x)} \rightsquigarrow N(0, \tau^2(x)) \quad \mathbf{H}$$

for some $\tau(x)$. This is true even if $h = h_n$ is decreasing. Specifically, suppose that $h_n \rightarrow 0$ and $nh_n \rightarrow \infty$. Note that $Z_n(x) = \sum_{i=1}^n L_{ni}$, say. According to Lyapounov's CLT, $\sum_{i=1}^n L_{ni} \rightsquigarrow N(0, 1)$ as long as

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n \mathbb{E}[L_{ni}]^{2+\delta} = 0$$

for some $\delta > 0$. But this does not yield a confidence interval for $p(x)$. To see why, let us write

$$\frac{\hat{p}_h(x) - p(x)}{s_n(x)} = \frac{\hat{p}_h(x) - p_h(x)}{s_n(x)} + \frac{p_h(x) - p(x)}{s_n(x)} = Z_n(x) + \frac{\text{bias}}{\sqrt{\text{var}(x)}}.$$

Assuming that the optimize the risk by balancing the bias and the variance, the second term is some constant c . So

$$\frac{\hat{p}_h(x) - p(x)}{s_n(x)} \rightsquigarrow N(c, \tau^2(x)).$$

This means that the usual confidence interval $\hat{p}_h(x) \pm z_{\alpha/2}s(x)$ will not cover $p(x)$ with probability tending to $1 - \alpha$. One fix for this is to undersmooth the estimator. (We sacrifice risk for coverage.) An easier approach is just to interpret $\hat{p}_h(x) \pm z_{\alpha/2}s(x)$ as a confidence interval for the smoothed density $p_h(x)$ instead of $p(x)$.

But this only gives an interval at one point. To get a confidence band we use the bootstrap. Let P_n be the empirical distribution of X_1, \dots, X_n . The idea is to estimate the distribution

$$F_n(t) = \mathbb{P}\left(\sqrt{nh^d}||\hat{p}_h(x) - p_h(x)||_\infty \leq t\right)$$

with the bootstrap estimator

$$\hat{F}_n(t) = \mathbb{P}\left(\sqrt{nh^d}||\hat{p}_h^*(x) - \hat{p}_h(x)||_\infty \leq t \mid X_1, \dots, X_n\right)$$

where \hat{p}_h^* is constructed from the bootstrap sample $X_1^*, \dots, X_n^* \sim P_n$. Later in the course, we will show that

$$\sup_t |F_n(t) - \hat{F}_n(t)| \xrightarrow{P} 0.$$

Here is the algorithm.

1. Let P_n be the empirical distribution that puts mass $1/n$ at each data point X_i .

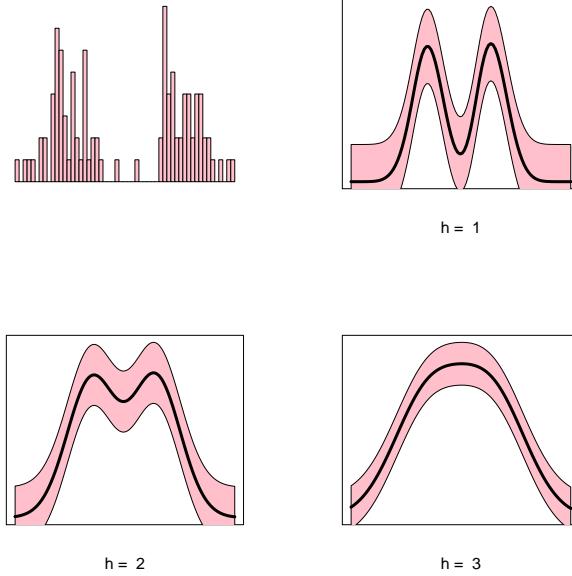


Figure 4: 95 percent bootstrap confidence bands using various bandwidths.

2. Draw $X_1^*, \dots, X_n^* \sim P_n$. This is called a bootstrap sample.
3. Compute the density estimator \hat{p}_h^* based on the bootstrap sample.
4. Compute $R = \sup_x \sqrt{nh^d} \|\hat{p}_h^* - \hat{p}_h\|_\infty$.
5. Repeat steps 2-4 B times. This gives R_1, \dots, R_B .
6. Let z_α be the upper α quantile of the R_j 's. Thus

$$\frac{1}{B} \sum_{j=1}^B I(R_j > z_\alpha) \approx \alpha.$$

7. Let

$$\ell_n(x) = \hat{p}_h(x) - \frac{z_\alpha}{\sqrt{nh^d}}, \quad u_n(x) = \hat{p}_h(x) + \frac{z_\alpha}{\sqrt{nh^d}}.$$

Theorem 10 *Under appropriate (very weak) conditions, we have*

$$\liminf_{n \rightarrow \infty} \mathbb{P}(\ell_n(x) \leq p_h(x) \leq u(x) \text{ for all } x) \geq 1 - \alpha.$$

See Figure 4.

If you want a confidence band for p you need to reduce the bias (undersmooth). A simple way to do this is with *twicing*. Suppose that $\beta = 2$ and that we use the kernel estimator \hat{p}_h . Note that,

$$\begin{aligned} \mathbb{E}[\hat{p}_h(x)] &= p(x) + C(x)h^2 + o(h^2) \\ \mathbb{E}[\hat{p}_{2h}(x)] &= p(x) + C(x)4h^2 + o(h^2) \end{aligned}$$

for some $C(x)$. That is, the leading term of the bias is $b(x) = C(x)h^2$. So if we define

$$\widehat{b}(x) = \frac{\widehat{p}_{2h}(x) - \widehat{p}_h(x)}{3}$$

then

$$\mathbb{E}[\widehat{b}(x)] = b(x).$$

We define the bias-reduced estimator

$$\widetilde{p}_h(x) = \widehat{p}_h(x) - \widehat{b}(x) = \frac{4}{3} \left(\widehat{p}_h(x) - \frac{1}{4} \widehat{p}_{2h} \right).$$

A confidence set centered at \widetilde{p}_h will be asymptotically valid but will not be an optimal estimator. This is a fundamental conflict between estimation and inference.

5 Cross-Validation

In practice we need a data-based method for choosing the bandwidth h . To do this, we will need to estimate the risk of the estimator and minimize the estimated risk over h . Here, we describe two cross-validation methods.

5.1 Leave One Out

A common method for estimating risk is leave-one-out cross-validation. Recall that the loss function is

$$\int (\widehat{p}(x) - p(x))^2 dx = \int \widehat{p}^2(x) dx - 2 \int \widehat{p}(x)p(x)dx + \int p^2(x)dx.$$

The last term does not involve \widehat{p} so we can drop it. Thus, we now define the loss to be

$$L(h) = \int \widehat{p}^2(x) dx - 2 \int \widehat{p}(x)p(x)dx.$$

The risk is $R(h) = \mathbb{E}(L(h))$.

Definition 11 *The leave-one-out cross-validation estimator of risk is*

$$\widehat{R}(h) = \int \left(\widehat{p}_{(-i)}(x) \right)^2 dx - \frac{2}{n} \sum_{i=1}^n \widehat{p}_{(-i)}(X_i) \tag{26}$$

where $\widehat{p}_{(-i)}$ is the density estimator obtained after removing the i^{th} observation.

It is easy to check that $\mathbb{E}[\widehat{R}(h)] = R(h)$.

When the kernel is Gaussian, the cross-validation score can be written, after some tedious algebra, as follows. Let $\phi(z; \sigma)$ denote a Normal density with mean 0 and variance σ^2 . Then,

$$\widehat{R}(h) = \frac{\phi^d(0; \sqrt{2}h)}{(n-1)} + \frac{n-2}{n(n-1)^2} \sum_{i \neq j} \prod_{\ell=1}^d \phi(X_{i\ell} - X_{j\ell}; \sqrt{2}h) \quad (27)$$

$$- \frac{2}{n(n-1)} \sum_{i \neq j} \prod_{\ell=1}^d \phi(X_{i\ell} - X_{j\ell}; h). \quad (28)$$

The estimator \widehat{p} and the cross-validation score can be computed quickly using the fast Fourier transform; see pages 61–66 of Silverman (1986).

For histograms, it is easy to work out the leave-one-out cross-validation in close form:

$$\widehat{R}(h) = \frac{2}{(n-1)h} - \frac{n+1}{(n-1)h} \sum_j \widehat{\theta}_j^2. \quad \text{H}$$

A further justification for cross-validation is given by the following theorem due to Stone (1984).

Theorem 12 (Stone's theorem) *Suppose that p is bounded. Let \widehat{p}_h denote the kernel estimator with bandwidth h and let \widehat{h} denote the bandwidth chosen by cross-validation. Then,*

$$\frac{\int (p(x) - \widehat{p}_{\widehat{h}}(x))^2 dx}{\inf_h \int (p(x) - \widehat{p}_h(x))^2 dx} \xrightarrow{a.s.} 1. \quad (29)$$

The bandwidth for the density estimator in the bottom left panel of Figure 1 is based on cross-validation. In this case it worked well but of course there are lots of examples where there are problems. Do not assume that, if the estimator \widehat{p} is wiggly, then cross-validation has let you down. The eye is not a good judge of risk.

There are cases when cross-validation can seriously break down. In particular, if there are ties in the data then cross-validation chooses a bandwidth of 0.

5.2 Data Splitting

An alternative to leave-one-out is V -fold cross-validation. A common choice is $V = 10$. For simplicity, let us consider here just splitting the data in two halves. This version of cross-validation comes with stronger theoretical guarantees. Let \widehat{p}_h denote the kernel estimator

based on bandwidth h . For simplicity, assume the sample size is even and denote the sample size by $2n$. Randomly split the data $X = (X_1, \dots, X_{2n})$ into two sets of size n . Denote these by $Y = (Y_1, \dots, Y_n)$ and $Z = (Z_1, \dots, Z_n)$.¹ Let $\mathcal{H} = \{h_1, \dots, h_N\}$ be a finite grid of bandwidths. Let

$$\hat{p}_j(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_j^d} K\left(\frac{\|x - Y_i\|}{h}\right).$$

Thus we have a set $\mathcal{P} = \{\hat{p}_1, \dots, \hat{p}_N\}$ of density estimators.

We would like to minimize $L(p, \hat{p}_j) = \int \hat{p}_j^2(x) - 2 \int \hat{p}_j(x)p(x)dx$. Define the estimated risk

$$\hat{L}_j \equiv \hat{L}(p, \hat{p}_j) = \int \hat{p}_j^2(x) - \frac{2}{n} \sum_{i=1}^n \hat{p}_j(Z_i). \quad (30)$$

Let $\hat{p} = \operatorname{argmin}_{g \in \mathcal{P}} \hat{L}(p, g)$. Schematically:

$$\begin{array}{ccc} & Y \rightarrow \{\hat{p}_1, \dots, \hat{p}_N\} = \mathcal{P} \\ X = (X_1, \dots, X_{2n}) & \xrightarrow{\text{split}} & Z \rightarrow \{\hat{L}_1, \dots, \hat{L}_N\} \end{array}$$

Theorem 13 (Wegkamp 1999) *There exists a $C > 0$ such that*

$$\mathbb{E}(\|\hat{p} - p\|^2) \leq 2 \min_{g \in \mathcal{P}} \mathbb{E}(\|g - p\|^2) + \frac{C \log N}{n}.$$

This theorem can be proved using concentration of measure techniques that we discuss later in class. A similar result can be proved for V -fold cross-validation.

5.3 Asymptotic Expansions

In this section we consider some asymptotic expansions that describe the behavior of the kernel estimator. We focus on the case $d = 1$.

Theorem 14 *Let $R_x = \mathbb{E}(p(x) - \hat{p}(x))^2$ and let $R = \int R_x dx$. Assume that p'' is absolutely continuous and that $\int p'''(x)^2 dx < \infty$. Then,*

$$R_x = \frac{1}{4} \sigma_K^4 h_n^4 p''(x)^2 + \frac{p(x) \int K^2(x) dx}{nh_n} + O\left(\frac{1}{n}\right) + O(h_n^6)$$

¹It is not necessary to split the data into two sets of equal size. We use the equal split version for simplicity.

and

$$R = \frac{1}{4}\sigma_K^4 h_n^4 \int p''(x)^2 dx + \frac{\int K^2(x) dx}{nh} + O\left(\frac{1}{n}\right) + O(h_n^6) \quad (31)$$

where $\sigma_K^2 = \int x^2 K(x) dx$.

Proof. Write $K_h(x, X) = h^{-1}K((x - X)/h)$ and $\hat{p}(x) = n^{-1} \sum_i K_h(x, X_i)$. Thus, $\mathbb{E}[\hat{p}(x)] = \mathbb{E}[K_h(x, X)]$ and $\text{Var}[\hat{p}(x)] = n^{-1}\text{Var}[K_h(x, X)]$. Now,

$$\begin{aligned} \mathbb{E}[K_h(x, X)] &= \int \frac{1}{h} K\left(\frac{x-t}{h}\right) p(t) dt \\ &= \int K(u)p(x-hu) du \\ &= \int K(u) \left[p(x) - h u p'(x) + \frac{h^2 u^2}{2} p''(x) + \dots \right] du \\ &= p(x) + \frac{1}{2} h^2 p''(x) \int u^2 K(u) du \dots \end{aligned}$$

since $\int K(x) dx = 1$ and $\int x K(x) dx = 0$. The bias is

$$\mathbb{E}[K_{h_n}(x, X)] - p(x) = \frac{1}{2}\sigma_K^2 h_n^2 p''(x) + O(h_n^4).$$

By a similar calculation,

$$\text{Var}[\hat{p}(x)] = \frac{p(x) \int K^2(x) dx}{n h_n} + O\left(\frac{1}{n}\right).$$

The first result then follows since the risk is the squared bias plus variance. The second result follows from integrating the first result. \square

If we differentiate (31) with respect to h and set it equal to 0, we see that the asymptotically optimal bandwidth is

$$h_* = \left(\frac{c_2}{c_1^2 A(f)n} \right)^{1/5} \quad (32)$$

where $c_1 = \int x^2 K(x) dx$, $c_2 = \int K(x)^2 dx$ and $A(f) = \int f''(x)^2 dx$. This is informative because it tells us that the best bandwidth decreases at rate $n^{-1/5}$. Plugging h_* into (31), we see that if the optimal bandwidth is used then $R = O(n^{-4/5})$.

6 High Dimensions

The rate of convergence $n^{-2\beta/(2\beta+d)}$ is slow when the dimension d is large. In this case it is hopeless to try to estimate the true density p precisely in the L_2 norm (or any similar norm).

We need to change our notion of what it means to estimate p in a high-dimensional problem. Instead of estimating p precisely we have to settle for finding an adequate approximation of p . Any estimator that finds the regions where p puts large amounts of mass should be considered an adequate approximation. Let us consider a few ways to implement this type of thinking.

Biased Density Estimation. Let $p_h(x) = \mathbb{E}(\hat{p}_h(x))$. Then

$$p_h(x) = \int \frac{1}{h^d} K\left(\frac{\|x - u\|}{h}\right) p(u) du$$

so that the mean of \hat{p}_h can be thought of as a smoothed version of p . Let $P_h(A) = \int_A p_h(u) du$ be the probability distribution corresponding to p_h . Then

$$P_h = P \oplus K_h$$

where \oplus denotes convolution² and K_h is the distribution with density $h^{-d}K(\|u\|/h)$. In other words, if $X \sim P_h$ then $X = Y + Z$ where $Y \sim P$ and $Z \sim K_h$. This is just another way to say that P_h is a blurred or smoothed version of P . p_h need not be close in L_2 to p but still could preserve most of the important shape information about p . Consider then choosing a fixed $h > 0$ and estimating p_h instead of p . This corresponds to ignoring the bias in the density estimator. From Theorem 8 we conclude:

Theorem 15 *Let $h > 0$ be fixed. Then $\mathbb{P}(\|\hat{p}_h - p_h\|_\infty > \epsilon) \leq Ce^{-n\epsilon^2}$. Hence,*

$$\|\hat{p}_h - p_h\|_\infty = O_P\left(\sqrt{\frac{\log n}{n}}\right).$$

The rate of convergence is fast and is independent of dimension. How to choose h is not clear.

Independence Based Methods. If we can live with some bias, we can reduce the dimensionality by imposing some independence assumptions. The simplest example is to treat the components (X_1, \dots, X_d) as if they are independent. In that case

$$p(x_1, \dots, x_d) = \prod_{j=1}^d p_j(x_j)$$

and the problem is reduced to a set of one-dimensional density estimation problems.

²If $X \sim P$ and $Y \sim Q$ are independent, then the distribution of $X + Y$ is denoted by $P \star Q$ and is called the convolution of P and Q .

An extension is to use a forest. We represent the distribution with an undirected graph. A graph with no cycles is a forest. Let E be the edges of the graph. Any density consistent with the forest can be written as

$$p(x) = \prod_{j=1}^d p_j(x_j) \prod_{(j,k) \in E} \frac{p_{j,k}(x_j, x_k)}{p_j(x_j)p_k(x_k)}.$$

To estimate the density therefore only require that we estimate one and two-dimensional marginals. But how do we find the edge set E ? Some methods are discussed in Liu et al (2011) under the name “Forest Density Estimation.” A simple approach is to connect pairs greedily using some measure of correlation.

Density Trees. Ram and Gray (2011) suggest a recursive partitioning scheme similar to decision trees. They split each coordinate dyadically, in a greedy fashion. The density estimator is taken to be piecewise constant. They use an L_2 risk estimator to decide when to split. This seems promising. The ideas seems to have been re-discovered in Yand and Wong (arXiv:1404.1425) and Liu and Wong (arXiv:1401.2597). Density trees seem very promising. It would be nice if there was an R package to do this and if there were more theoretical results.

7 Example

Figure 5 shows a synthetic two-dimensional data set, the cross-validation function and two kernel density estimators. The data are 100 points generated as follows. We select a point randomly on the unit circle then add Normal noise with standard deviation 0.1. The first estimator (lower left) uses the bandwidth that minimizes the leave-one-out cross-validation score. The second uses twice that bandwidth. The cross-validation curve is very sharply peaked with a clear minimum. The resulting density estimate is somewhat lumpy. This is because cross-validation is aiming to minimize L_2 error which does not guarantee that the estimate is smooth. Also, the dataset is small so this effect is more noticeable. The estimator with the larger bandwidth is noticeably smoother. However, the lumpiness of the estimator is not necessarily a bad thing.

8 Derivatives

Kernel estimators can also be used to estimate the derivatives of a density.³ Let $D^{\otimes r}p$ denote the r^{th} derivative p . We are using Kronecker notation. Let $D^{\otimes 0}p = p$, $D^{\otimes 1}f$ is the gradient

³In this section we follow Chacon and Duong (2013), *Electronic Journal of Statistics*, 7, 499-532.

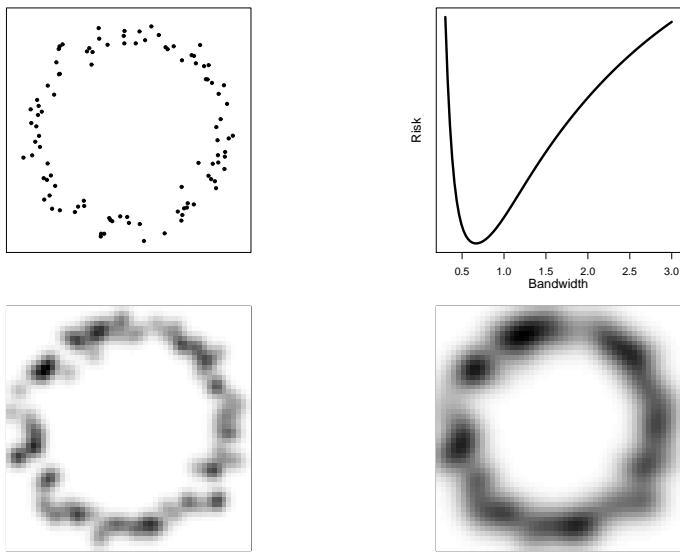


Figure 5: Synthetic two-dimensional data set. Top left: data. Top right: cross-validation function. Bottom left: kernel estimator based on the bandwidth that minimizes the cross-validation score. Bottom right: kernel estimator based on twice the bandwidth that minimizes the cross-validation score.

of p , and $D^{\otimes 2}p = \text{vec}\mathcal{H}$ where \mathcal{H} is the Hessian. We also write this as $p^{(r)}$ when convenient.

Let H be a bandwidth matrix and let

$$\widehat{p}_H(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - X_i)$$

where $K_H(x) = |H|^{-1/2}K(H^{-1/2}x)$. We define

$$\widehat{p}^{(r)}(x) = D^{\otimes r} \widehat{p}_H(x) = \frac{1}{n} \sum_{i=1}^n D^{\otimes r} K_H(x - X_i).$$

For computation, it is useful to note that

$$D^{\otimes r} K_H(x) = |H|^{-1/2} (H^{-1/2})^{\otimes r} D^{\otimes r} K_H(H^{-1/2}x).$$

The asymptotic mean squared error is derived in Chacon, Duong and Wand (2011) and is given by

$$\frac{1}{n} |H|^{-1/2} \text{tr}((H^{-1})^{\otimes r} R(D^{\otimes r}(K))) + \frac{m_2^2(K)}{4} \text{tr}((I_{d^r} \otimes \text{vec}^T H) R(D^{\otimes(r+2)} p)(I_{d^r} \otimes \text{vec}(H)))$$

where $R(g) = \int g(x)g^T(x)dx$, $m_2(K) = \int xx^T K(x)dx$. The optimal H has entries of order $n^{-2/(d+2r+4)}$ which yield an asymptotic mean squared error of order $n^{-4/(d+2r+4)}$. In dimension $d = 1$, the risk looks like this as a function of r :

r	risk
0	$n^{-4/5}$
1	$n^{-4/7}$
2	$n^{-4/9}$

We see that estimating derivatives is harder than estimating the density itself.

Chacon and Duong (2013) derive an estimate of the risk:

$$\text{CV}_r(H) = (-1)^r |H|^{-1/2} \text{vec}^T (H^{-1})^{\otimes r} G_n$$

where

$$G_n = \frac{1}{n^2} \sum_{i,j} D^{\otimes 2r} \overline{K}(H^{-1/2}(X_i - X_j)) - \frac{2}{n(n-1)} \sum_{i \neq j} D^{\otimes 2r} K(H^{-1/2}(X_i - X_j))$$

and $\overline{K} = K \star K$. We can now minimize CV over H . It would be nice if someone wrote an R package to do this. I think the ks package does much of this.

One application of this that we consider later in the course is mode-based clustering. Here, we use density estimation to find the modes of the density. We associate clusters with these modes. We can also test for a mode by testing if $D^2p(x) < 0$ at the estimated modes.

9 Unsupervised Prediction and Anomaly Detection

We can use density estimation to do unsupervised prediction and anomaly detection. The basic idea is due to Vovk, and was developed in a statistical framework in Lei, Robins and Wasserman (2014).

Suppose we observe $Y_1, \dots, Y_n \sim P$. We want to predict Y_{n+1} . We will construct a level α test for the null hypothesis $H_0 : Y_{n+1} = y$. We do this for every value of y . Then we invert the test, that is, we set C_n to be the set of y 's that are not rejected. It follows that

$$\mathbb{P}(Y_{n+1} \in C_n) \geq 1 - \alpha.$$

The prediction set C_n is finite sample and distribution-free.

Fix a value y . Let $A = (Y_1, \dots, Y_n, y)$ be the *augmented dataset*. That is, we set $Y_{n+1} = y$. Let \hat{p}_A be a density estimate based on A . Consider the vector

$$\hat{p}_A(Y_1), \dots, \hat{p}_A(Y_{n+1}).$$

Under H_0 , the rank of these values is uniformly distributed. That is, for each i ,

$$\mathbb{P}(\hat{p}_A(Y_i) \leq \hat{p}_A(y)) = \frac{1}{n+1}.$$

A p-value for the test is

$$\pi(y) = \frac{1}{n+1} \sum_{i=1}^{n+1} I(\hat{p}_A(Y_i) \leq \hat{p}_A(y)).$$

The prediction set is

$$C_n = \left\{ y : \pi(y) \geq \alpha \right\}.$$

Computing C_n is tedious. Fortunately, Jing, Robins and Wasserman (2014) show that there is a simpler set that still has the correct coverage (but is slightly larger). The set is constructed as follows. Let $Z_i = \hat{p}(Y_i)$. Order these observations

$$Z_{(1)} \leq \dots \leq Z_{(n)}.$$

Let $k = \lfloor (n+1)\alpha \rfloor$ and let

$$t = Z_{(k)} - \frac{K(0)}{nh^d}.$$

Define

$$C_n^+ = \left\{ y : \hat{p}(y) \geq t \right\}.$$

Lemma 16 We have that $C_n \subset C_n^+$ and hence

$$\mathbb{P}(Y_{n+1} \in C_n) \geq 1 - \alpha.$$

Finally, we note that any Y_i with a small p-value can be regarded as an outlier (anomaly).

The above method is exact. We can also use a simpler, asymptotic approach. With $Z_{(k)}$ defined above, set $\widehat{C} = \{y : \widehat{p}(y) \geq t\}$ where now $t = Z_{(k)}$. From Cadre, Pelletier and Pudlo (2013) we have that

$$\sqrt{nh^d} \mu(\widehat{C} \Delta C) \xrightarrow{P} c$$

for some constant c where C is the true $1 - \alpha$ level set. Hence, $P(Y_{n+1} \in \widehat{C}) = 1 - \alpha + o_P(1)$.

10 Manifolds and Singularities

Sometimes a distribution is concentrated near a lower-dimensional set. This causes problems for density estimation. In fact the density, as we usually think of it, may not be defined.

As a simple example, suppose P is supported on the unit circle in \mathbb{R}^2 . The distribution P is *singular* with respect to Lebesgue measure μ . This means that there are sets A with $P(A) > 0$ even though $\mu(A) = 0$. Effectively, this means that the density is infinite. To see this, consider a point x on the circle. Let $B(x, \epsilon)$ be a ball of radius ϵ centered at x . Then

$$p(x) = \lim_{\epsilon \rightarrow 0} \frac{\mathbb{P}(B(x, \epsilon))}{\mu(B(x, \epsilon))} \rightarrow \infty.$$

H

Note also that the L_2 loss does not make any sense. If you tried to use cross-validation, you would find that the estimated risk is minimized at $h = 0$.

H

A simple solution is to focus on estimating the smoothed density $p_h(x)$ which is well-defined for every $h > 0$. More sophisticated ideas are based on topological data analysis which we discuss later in the course.

11 Series Methods

We have emphasized kernel density estimation. There are many other density estimation methods. Let us briefly mention a method based on basis functions. For simplicity, suppose that $X_i \in [0, 1]$ and let ϕ_1, ϕ_2, \dots be an orthonormal basis for

$$\mathcal{F} = \{f : [0, 1] \rightarrow \mathbb{R}, \int_0^1 f^2(x) dx < \infty\}.$$

Thus

$$\int \phi_j^2(x) dx = 1, \quad \int \phi_j(x) \phi_k(x) dx = 0.$$

An example is the cosine basis:

$$\phi_0(x) = 1, \quad \phi_j(x) = \sqrt{2} \cos(2\pi jx), \quad j = 1, 2, \dots,$$

If $p \in \mathcal{F}$ then

$$p(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$$

where $\beta_j = \int_0^1 p(x) \phi_j(x) dx$. An estimate of p is $\hat{p}(x) = \sum_{j=1}^k \hat{\beta}_j \phi_j(x)$ where

$$\hat{\beta}_j = \frac{1}{n} \sum_{i=1}^n \phi_j(X_i).$$

The number of terms k is the smoothing parameter and can be chosen using cross-validation.

It can be shown that

$$R = \mathbb{E}[\int (\hat{p}(x) - p(x))^2 dx] = \sum_{j=1}^n \text{Var}(\hat{\beta}_j) + \sum_{j=k+1}^{\infty} \beta_j^2. \quad \mathbf{H}$$

The first term is of order $O(k/n)$. To bound the second term (the bias) one usually assumes that p is a *Sobolev space of order q* which means that $p \in \mathcal{P}$ with

$$\mathcal{P} = \left\{ p \in \mathcal{F} : p = \sum_j \beta_j \phi_j : \sum_{j=1}^{\infty} \beta_j^2 j^{2q} < \infty \right\}.$$

In that case it can be shown that

$$R \approx \frac{k}{n} + \left(\frac{1}{k} \right)^{2q}. \quad \mathbf{H}$$

The optimal k is $k \approx n^{1/(2q+1)}$ with risk

$$R = O \left(\frac{1}{n} \right)^{\frac{2q}{2q+1}}.$$

11.1 L_1 Methods

Here we discuss another approach to choosing h aimed at the L_1 loss. The idea is to select a class of sets \mathcal{A} —which we call test sets—and choose h to make $\int_A \hat{p}_h(x) dx$ close to $P(A)$ for all $A \in \mathcal{A}$. That is, we would like to minimize

$$\Delta(g) = \sup_{A \in \mathcal{A}} \left| \int_A g(x) dx - P(A) \right|. \quad (33)$$

VC Classes. Let \mathcal{A} be a class of sets with VC dimension ν . As in section 5.2, split the data X into Y and Z with $\mathcal{P} = \{\hat{p}_1, \dots, \hat{p}_N\}$ constructed from Y . For $g \in \mathcal{P}$ define

$$\Delta_n(g) = \sup_{A \in \mathcal{A}} \left| \int_A g(x) dx - P_n(A) \right|$$

where $P_n(A) = n^{-1} \sum_{i=1}^n I(Z_i \in A)$. Let $\hat{p} = \operatorname{argmin}_{g \in \mathcal{P}} \Delta_n(g)$.

Theorem 17 *For any $\delta > 0$ there exists c such that*

$$\mathbb{P} \left(\Delta(\hat{p}) > \min_j \Delta(\hat{p}_j) + 2c \sqrt{\frac{\nu}{n}} \right) < \delta.$$

Proof. We know that

$$\mathbb{P} \left(\sup_{A \in \mathcal{A}} |P_n(A) - P(A)| > c \sqrt{\frac{\nu}{n}} \right) < \delta.$$

Hence, except on an event of probability at most δ , we have that

$$\begin{aligned} \Delta_n(g) &= \sup_{A \in \mathcal{A}} \left| \int_A g(x) dx - P_n(A) \right| \leq \sup_{A \in \mathcal{A}} \left| \int_A g(x) dx - P(A) \right| + \sup_{A \in \mathcal{A}} \left| P_n(A) - P(A) \right| \\ &\leq \Delta(g) + c \sqrt{\frac{\nu}{n}}. \end{aligned}$$

By a similar argument, $\Delta(g) \leq \Delta_n(g) + c \sqrt{\frac{\nu}{n}}$. Hence, $|\Delta(g) - \Delta_n(g)| \leq c \sqrt{\frac{\nu}{n}}$ for all g . Let $p_* = \operatorname{argmin}_{g \in \mathcal{P}} \Delta(g)$. Then,

$$\Delta(p) \leq \Delta(\hat{p}) \leq \Delta_n(\hat{p}) + c \sqrt{\frac{\nu}{n}} \leq \Delta_n(p_*) + c \sqrt{\frac{\nu}{n}} \leq \Delta(p_*) + 2c \sqrt{\frac{\nu}{n}}.$$

□

The difficulty in implementing this idea is computing and minimizing $\Delta_n(g)$. Hjort and Walker (2001) presented a similar method which can be practically implemented when $d = 1$.

Yatracos Classes. Devroye and Györfi (2001) use a class of sets called a Yatracos class which leads to estimators with some remarkable properties. Let $\mathcal{P} = \{p_1, \dots, p_N\}$ be a set of densities and define the Yatracos class of sets $\mathcal{A} = \{A(i, j) : i \neq j\}$ where $A(i, j) = \{x : p_i(x) > p_j(x)\}$. Let

$$\hat{p} = \operatorname{argmin}_{g \in \mathcal{G}} \Delta(g)$$

where

$$\Delta_n(g) = \sup_{A \in \mathcal{A}} \left| \int_A g(u) du - P_n(A) \right|$$

and $P_n(A) = n^{-1} \sum_{i=1}^n I(Z_i \in A)$ is the empirical measure based on a sample $Z_1, \dots, Z_n \sim p$.

Theorem 18 *The estimator \hat{p} satisfies*

$$\int |\hat{p} - p| \leq 3 \min_j \int |p_j - p| + 4\Delta \quad (34)$$

where $\Delta = \sup_{A \in \mathcal{A}} \left| \int_A p - P_n(A) \right|$.

The term $\min_j \int |p_j - p|$ is like a bias while term Δ is like the variance.

Proof. Let i be such that $\hat{p} = p_i$ and let s be such that $\int |p_s - p| = \min_j \int |p_j - p|$. Let $B = \{p_i > p_s\}$ and $C = \{p_s > p_i\}$. Now,

$$\int |\hat{p} - p| \leq \int |p_s - p| + \int |p_s - p_i|. \quad (35)$$

Let \mathcal{B} denote all measurable sets. Then,

$$\begin{aligned} \int |p_s - p_i| &= 2 \max_{A \in \{B, C\}} \left| \int_A p_i - \int_A p_s \right| \leq 2 \sup_{A \in \mathcal{A}} \left| \int_A p_i - \int_A p_s \right| \\ &\leq 2 \sup_{A \in \mathcal{A}} \left| \int_A p_i - P_n(A) \right| + 2 \sup_{A \in \mathcal{A}} \left| \int_A p_s - P_n(A) \right| \\ &\leq 4 \sup_{A \in \mathcal{A}} \left| \int_A p_s - P_n(A) \right| \\ &\leq 4 \sup_{A \in \mathcal{A}} \left| \int_A p_s - \int_A p \right| + 4 \sup_{A \in \mathcal{A}} \left| \int_A p - P_n(A) \right| \\ &= 4 \sup_{A \in \mathcal{A}} \left| \int_A p_s - \int_A p \right| + 4\Delta \leq 4 \sup_{A \in \mathcal{B}} \left| \int_A p_s - \int_A p \right| + 4\Delta \\ &= 2 \int |p_s - p| + 4\Delta. \end{aligned}$$

The result follows from (35). \square

Now we apply this to kernel estimators. Again we split the data X into two halves $Y = (Y_1, \dots, Y_n)$ and $Z = (Z_1, \dots, Z_n)$. For each h let

$$\hat{p}_h(x) = \frac{1}{n} \sum_{i=1}^n K \left(\frac{\|x - Y_i\|}{h} \right).$$

Let

$$\mathcal{A} = \left\{ A(h, \nu) : h, \nu > 0, h \neq \nu \right\}$$

where $A(h, \nu) = \{x : \hat{p}_h(x) > \hat{p}_\nu(x)\}$. Define

$$\Delta_n(g) = \sup_{A \in \mathcal{A}} \left| \int_A g(u) du - P_n(A) \right|$$

where $P_n(A) = n^{-1} \sum_{i=1}^n I(Z_i \in A)$ is the empirical measure based on Z . Let

$$\hat{p} = \operatorname{argmin}_{g \in \mathcal{G}} \Delta(g).$$

Under some regularity conditions on the kernel, we have the following result.

Theorem 19 (*Devroye and Györfi, 2001.*) *The risk of \hat{p} satisfies*

$$\mathbb{E} \int |\hat{p} - p| \leq c_1 \inf_h \mathbb{E} \int |\hat{p}_h - p| + c_2 \sqrt{\frac{\log n}{n}}. \quad (36)$$

The proof involves showing that the terms on the right hand side of (34) are small. We refer the reader to Devroye and Györfi (2001) for the details.

Recall that $d_{TV}(P, Q) = \sup_A |P(A) - Q(A)| = (1/2) \int |p(x) - q(x)| dx$ where the supremum is over all measurable sets. The above theorem says that the estimator does well in the total variation metric, even though the method only used the Yatracos class of sets. Finding computationally efficient methods to implement this approach remains an open question.

12 Mixtures

Another approach to density estimation is to use mixtures. We will discuss mixture modelling when we discuss clustering.

13 Two-Sample Hypothesis Testing

Density estimation can be used for two sample testing. Given $X_1, \dots, X_n \sim p$ and $Y_1, \dots, Y_m \sim q$ we can test $H_0 : p = q$ using $\int (\hat{p} - \hat{q})^2$ as a test statistic. More interestingly, we can test locally $H_0 : p(x) = q(x)$ at each x . See Duong (2013) and Kim, Lee and Lei (2018). Note that under H_0 , the bias cancels from $\hat{p}(x) - \hat{q}(x)$. Also, some sort of multiple testing correction is required.

14 Functional Data and Quasi-Densities

In some problems, X is not just high dimensional, it is infinite dimensional. For example suppose that each X_i is a curve. An immediate problem is that the concept of a density is no longer well defined. On a Euclidean space, the density p for a probability measure is

the function that satisfies $P(A) = \int_A p(u)d\mu(u)$ for all measurable A where μ is Lebesgue measure. Formally, we say that p is the Radon-Nikodym derivative of P with respect to the dominating measure μ . Geometrically, we can think of p as

$$p(x) = \lim_{\epsilon \rightarrow 0} \frac{\mathbb{P}(\|X - x\| \leq \epsilon)}{V(\epsilon)}$$

where $V(\epsilon) = \epsilon^d \pi^{d/2} / \Gamma(d/2 + 1)$ is the volume of a sphere of radius ϵ . Under appropriate conditions, these two notions of density agree. (This is the Lebesgue density theorem.)

When the outcome space \mathcal{X} is a set of curves, there is no dominating measure and hence there is no density. Instead, we define the density geometrically by

$$q_\epsilon(x) = \mathbb{P}(\xi(x, X) \leq \epsilon)$$

for a small ϵ where ξ is some metric on \mathcal{X} . However we cannot divide by $V(\epsilon)$ and let ϵ tend to 0 since the dimension d is infinite.

One way around this is to use a fixed ϵ and work with the unnormalized density q_ϵ . For the purpose of finding high-density regions this may be adequate. An estimate of q_ϵ is

$$\hat{q}_\epsilon(x) = \frac{1}{n} \sum_{i=1}^n I(\xi(x, X_i) \leq \epsilon).$$

An alternative is to expand X_i into a basis: $X(t) \approx \sum_{j=1}^k \beta_j \psi_j(t)$. A density can be defined in terms of the β_j 's.

Example 20 Figure 6 shows the tracks (or paths) of 40 North Atlantic tropical cyclones (TC). The full dataset, consisting of 608 from 1950 to 2006 is shown in Figure 7. Buchman, Lee and Schafer (2009) provide a thorough analysis of the data. We refer the reader to their paper for the full details.⁴

Each data point—that is, each TC track—is a curve in \mathbb{R}^2 . Various questions are of interest: Where are the tracks most common? Is the density of tracks changing over time? Is the track related to other variables such as windspeed and pressure?

Each curve X_i can be regarded as mapping $X_i : [0, T_i] \rightarrow \mathbb{R}^2$ where $X_i(t) = (X_{i1}(t), X_{i2}(t))$ is the position of the TC at time t and T_i is the lifelength of the TC. Let

$$\Gamma_i = \left\{ (X_{i1}(t), X_{i2}(t)) : 0 \leq t \leq T_i \right\}$$

be the graph of X_i . In other words, Γ_i is the track, regarded as a subset of points in \mathbb{R}^2 . We will use the Hausdorff metric to measure the distance between curves. The Hausdorff

⁴Thanks to Susan Buchman for providing the data.



Figure 6: Paths of 40 tropical cyclones in the North Atlantic.

distance between two sets A is B is

$$d_H(A, B) = \inf\{\epsilon : A \subset B^\epsilon \text{ and } B \subset A^\epsilon\} \quad (37)$$

$$= \max \left\{ \sup_{x \in A} \inf_{y \in B} \|x - y\|, \sup_{x \in B} \inf_{y \in A} \|x - y\| \right\} \quad (38)$$

where $A^\epsilon = \bigcup_{x \in A} B(x, \epsilon)$ is called the enlargement of A and $B(x, \epsilon) = \{y : \|y - x\| \leq \epsilon\}$. We use the unnormalized kernel estimator

$$\hat{q}_\epsilon(\gamma) = \frac{1}{n} \sum_{i=1}^n I(d_H(\gamma, \Gamma_i) \leq \epsilon).$$

Figure 8 shows the 10 TC's with highest local density and the the 10 TC's with lowest local density using $\epsilon = 16.38$. This choice of ϵ corresponds to the 10th percentile of the values $\{d_H(X_i, X_j) : i \neq j\}$. The high density tracks correspond to TC's in the gulf of Mexico with short paths. The low density tracks correspond to TC's in the Atlantic with long paths.

15 Miscellanea

Another method for selecting h which is sometimes used when p is thought to be very smooth is the plug-in method. The idea is to take the formula for the mean squared error (equation 31), insert a guess of p'' and then solve for the optimal bandwidth h . For example, if $d = 1$

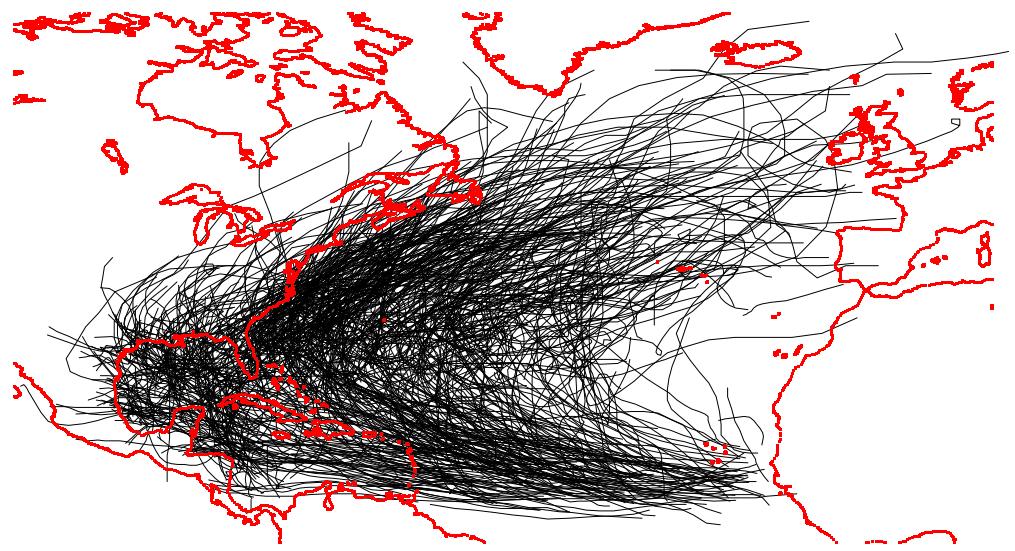


Figure 7: Paths of 608 tropical cyclones in the North Atlantic.

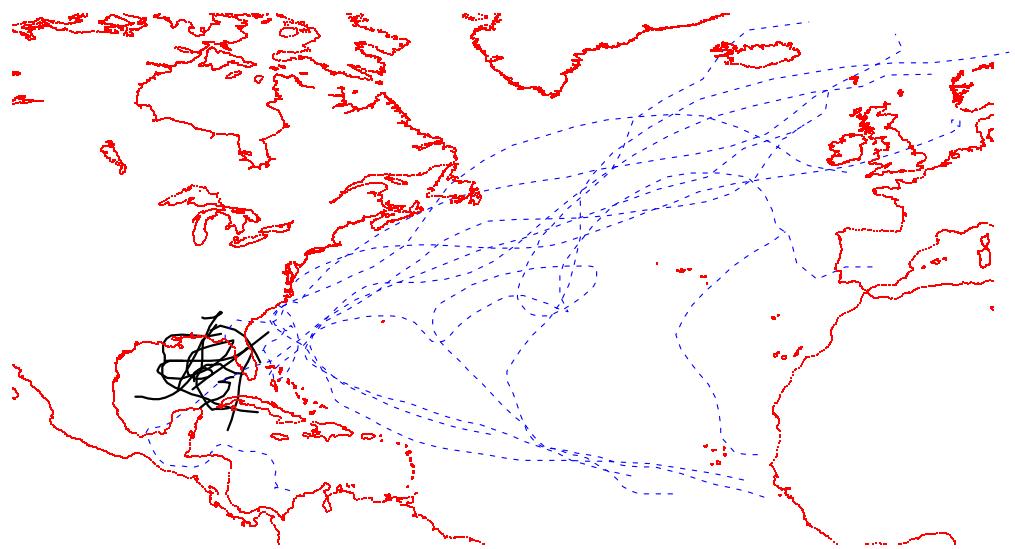


Figure 8: 10 highest density paths (black) and 10 lowest density paths (blue).

and under the idealized assumption that p is a univariate Normal this yields $h_* = 1.06\sigma n^{-1/5}$. Usually, σ is estimated by $\min\{s, Q/1.34\}$ where s is the sample standard deviation and Q is the interquartile range.⁵ This choice of h_* works well if the true density is very smooth and is called the Normal reference rule.

Since we don't want to necessarily assume that p is very smooth, it is usually better to estimate h using cross-validation. See Loader (1999) for an interesting comparison between cross-validation and plugin methods.

A generalization of the kernel method is to use adaptive kernels where one uses a different bandwidth $h(x)$ for each point x . One can also use a different bandwidth $h(x_i)$ for each data point. This makes the estimator more flexible and allows it to adapt to regions of varying smoothness. But now we have the very difficult task of choosing many bandwidths instead of just one.

Density estimation is sometimes used to find unusual observations or outliers. These are observations for which $\widehat{p}(X_i)$ is very small.

16 Summary

1. A commonly used nonparametric density estimator is the kernel estimator

$$\widehat{p}_h(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^d} K\left(\frac{\|x - X_i\|}{h}\right).$$

2. The kernel estimator is rate minimax over certain classes of densities.
3. Cross-validation methods can be used for choosing the bandwidth h .

⁵Recall that the interquartile range is the 75th percentile minus the 25th percentile. The reason for dividing by 1.34 is that $Q/1.34$ is a consistent estimate of σ if the data are from a $N(\mu, \sigma^2)$.

Nonparametric Regression

Statistical Machine Learning, Spring 2019
Ryan Tibshirani and Larry Wasserman

1 Introduction

1.1 Basic setup

Given a random pair $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$, recall that the function

$$m_0(x) = \mathbb{E}(Y|X = x)$$

is called the regression function (of Y on X). The basic goal in nonparametric regression: to construct a predictor of Y given X . This is basically the same as constructing an estimate \hat{m} of m_0 , from i.i.d. samples $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \dots, n$. Given a new X , our prediction of Y is $\hat{m}(X)$. We often call X the input, predictor, feature, etc., and Y the output, outcome, response, etc.

Note for i.i.d. samples $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \dots, n$, we can always write

$$Y_i = m_0(X_i) + \epsilon_i, \quad i = 1, \dots, n,$$

where ϵ_i , $i = 1, \dots, n$ are i.i.d. random errors, with mean zero. Therefore we can think about the sampling distribution as follows: (X_i, ϵ_i) , $i = 1, \dots, n$ are i.i.d. draws from some common joint distribution, where $\mathbb{E}(\epsilon_i) = 0$, and Y_i , $i = 1, \dots, n$ are generated from the above model.

It is common to assume that each ϵ_i is independent of X_i . This is a very strong assumption, and you should think about it skeptically. We too will sometimes make this assumption, for simplicity. It should be noted that a good portion of theoretical results that we cover (or at least, similar theory) also holds without this assumption.

1.2 Fixed or random inputs?

Another common setup in nonparametric regression is to directly assume a model

$$Y_i = m_0(X_i) + \epsilon_i, \quad i = 1, \dots, n,$$

where now X_i , $i = 1, \dots, n$ are *fixed* inputs, and ϵ_i , $i = 1, \dots, n$ are i.i.d. with $\mathbb{E}(\epsilon_i) = 0$.

For arbitrary X_i , $i = 1, \dots, n$, this is really just the same as starting with the random input model, and conditioning on the particular values of X_i , $i = 1, \dots, n$. (But note: after conditioning on the inputs, the errors are only i.i.d. if we assumed that the errors and inputs were independent in the first place.)

Generally speaking, nonparametric regression estimators are not defined with the random or fixed setups specifically in mind, i.e., there is no real distinction made here. A caveat: some estimators (like wavelets) do in fact assume evenly spaced fixed inputs, as in

$$X_i = i/n, \quad i = 1, \dots, n,$$

for evenly spaced inputs in the univariate case.

Theory is not completely the same between the random and fixed input worlds (some theory is sharper when we assume fixed input points, especially evenly spaced input points), but for the most part the theory is quite similar.

Therefore, in what follows, we won't be very precise about which setup we assume—random or fixed inputs—because it mostly doesn't matter when introducing nonparametric regression estimators and discussing basic properties.

1.3 Notation

We will define an empirical norm $\|\cdot\|_n$ in terms of the training points X_i , $i = 1, \dots, n$, acting on functions $m : \mathbb{R}^d \rightarrow \mathbb{R}$, by

$$\|m\|_n^2 = \frac{1}{n} \sum_{i=1}^n m^2(X_i).$$

This makes sense no matter if the inputs are fixed or random (but in the latter case, it is a random norm)

When the inputs are considered random, we will write P_X for the distribution of X , and we will define the L_2 norm $\|\cdot\|_2$ in terms of P_X , acting on functions $m : \mathbb{R}^d \rightarrow \mathbb{R}$, by

$$\|m\|_2^2 = \mathbb{E}[m^2(X)] = \int m^2(x) dP_X(x).$$

So when you see $\|\cdot\|_2$ in use, it is a hint that the inputs are being treated as random

A quantity of interest will be the (squared) error associated with an estimator \hat{m} of m_0 , which can be measured in either norm:

$$\|\hat{m} - m_0\|_n^2 \quad \text{or} \quad \|\hat{m} - m_0\|_2^2.$$

In either case, this is a random quantity (since \hat{m} is itself random). We will study bounds in probability or in expectation. The expectation of the errors defined above, in terms of either norm (but more typically the L_2 norm) is most properly called the risk; but we will often be a bit loose in terms of our terminology and just call this the error.

1.4 Bias-Variance Tradeoff

If (X, Y) is a new pair then

$$\mathbb{E}(Y - \hat{m}(X))^2 = \int b_n^2(x) dP(x) + \int v(x) dP(x) + \tau^2 = \|\hat{m} - m_0\|_2^2 + \tau^2$$

where $b_n(x) = \mathbb{E}[\hat{m}(x)] - m(x)$ is the bias, $v(x) = \text{Var}(\hat{m}(x))$ is the variance and $\tau^2 = \mathbb{E}(Y - m(X))^2$ is the un-avoidable error. Generally, we have to choose tuning parameters carefully to balance the bias and variance.

1.5 What does “nonparametric” mean?

Importantly, in nonparametric regression we don’t assume a particular parametric form for m_0 . This doesn’t mean, however, that we can’t estimate m_0 using (say) a linear combination of spline basis functions, written as $\hat{m}(x) = \sum_{j=1}^p \hat{\beta}_j g_j(x)$. A common question: the coefficients on the spline basis functions β_1, \dots, β_p are parameters, so how can this be nonparametric? Again, the point is that *we don’t assume a parametric form for m_0* , i.e., we don’t assume that m_0 itself is an exact linear combination of splines basis functions g_1, \dots, g_p .

1.6 What we cover here

The goal is to expose you to a variety of methods, and give you a flavor of some interesting results, under different assumptions. A few topics we will cover into more depth than others, but overall, this will be far from a complete treatment of nonparametric regression. Below are some excellent texts out there that you can consult for more details, proofs, etc.

Nearest neighbors. Kernel smoothing, local polynomials: [Tsybakov \(2009\)](#) Smoothing splines: [de Boor \(1978\)](#), [Green & Silverman \(1994\)](#), [Wahba \(1990\)](#) Reproducing kernel Hilbert spaces: [Scholkopf & Smola \(2002\)](#), [Wahba \(1990\)](#) Wavelets: [Johnstone \(2011\)](#), [Mallat \(2008\)](#). General references, more theoretical: [Gyorfi, Kohler, Krzyzak & Walk \(2002\)](#), [Wasserman \(2006\)](#) General references, more methodological: [Hastie & Tibshirani \(1990\)](#), [Hastie, Tibshirani & Friedman \(2009\)](#), [Simonoff \(1996\)](#)

Throughout, our discussion will bounce back and forth between the multivariate case ($d > 1$) and univariate case ($d = 1$). Some methods have obvious (natural) multivariate extensions; some don’t. In any case, we can always use low-dimensional (even just univariate) nonparametric regression methods as building blocks for a high-dimensional nonparametric method. We’ll study this near the end, when we talk about additive models.

1.7 Holder Spaces and Sobolev Spaces

The class of Lipschitz functions $H(1, L)$ on $T \subset \mathbb{R}$ is the set of functions g such that

$$|g(y) - g(x)| \leq L|x - y| \text{ for all } x, y \in T.$$

A differentiable function is Lipschitz if and only if it has bounded derivative. Conversely a Lipschitz function is differentiable almost everywhere.

Let $T \subset \mathbb{R}$ and let β be an integer. The Holder space $H(\beta, L)$ is the set of functions g mapping T to \mathbb{R} such that g is $\ell = \beta - 1$ times differentiable and satisfies

$$|g^{(\ell)}(y) - g^{(\ell)}(x)| \leq L|x - y| \text{ for all } x, y \in T.$$

(There is an extension to real valued β but we will not need that.) If $g \in H(\beta, L)$ and $\ell = \beta - 1$, then we can define the Taylor approximation of g at x by

$$\tilde{g}(y) = g(y) + (y - x)g'(x) + \dots + \frac{(y - x)^\ell}{\ell!}g^{(\ell)}(x)$$

and then $|g(y) - \tilde{g}(y)| \leq |y - x|^\beta$.

The definition for higher dimensions is similar. Let \mathcal{X} be a compact subset of \mathbb{R}^d . Let β and L be positive numbers. Given a vector $s = (s_1, \dots, s_d)$, define $|s| = s_1 + \dots + s_d$, $s! = s_1! \cdots s_d!$, $x^s = x_1^{s_1} \cdots x_d^{s_d}$ and

$$D^s = \frac{\partial^{s_1+\dots+s_d}}{\partial x_1^{s_1} \cdots \partial x_d^{s_d}}.$$

Let β be a positive integer. Define the *Hölder class*

$$H_d(\beta, L) = \left\{ g : |D^s g(x) - D^s g(y)| \leq L \|x - y\|, \text{ for all } s \text{ such that } |s| = \beta - 1, \text{ and all } x, y \right\}. \quad (1)$$

For example, if $d = 1$ and $\beta = 2$ this means that

$$|g'(x) - g'(y)| \leq L |x - y|, \quad \text{for all } x, y.$$

The most common case is $\beta = 2$; roughly speaking, this means that the functions have bounded second derivatives.

Again, if $g \in H_d(\beta, L)$ then $g(x)$ is close to its Taylor series approximation:

$$|g(u) - g_{x,\beta}(u)| \leq L \|u - x\|^\beta \quad (2)$$

where

$$g_{x,\beta}(u) = \sum_{|s| \leq \beta} \frac{(u - x)^s}{s!} D^s g(x). \quad (3)$$

In the common case of $\beta = 2$, this means that

$$\left| p(u) - [p(x) + (x - u)^T \nabla p(x)] \right| \leq L \|x - u\|^2.$$

The Sobolev class $S_1(\beta, L)$ is the set of β times differentiable functions (technically, it only requires weak derivatives) $g : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$\int (g^{(\beta)}(x))^2 dx \leq L^2.$$

Again this extends naturally to \mathbb{R}^d . Also, there is an extension to non-integer β . It can be shown that $H_d(\beta, L) \subset S_d(\beta, L)$.

2 *k*-nearest-neighbors regression

Here's a basic method to start us off: *k*-nearest-neighbors regression. We fix an integer $k \geq 1$ and define

$$\hat{m}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} Y_i, \quad (4)$$

where $\mathcal{N}_k(x)$ contains the indices of the k closest points of X_1, \dots, X_n to x .

This is not at all a bad estimator, and you will find it used in lots of applications, in many cases probably because of its simplicity. By varying the number of neighbors k , we can achieve a wide range of flexibility in the estimated function \hat{m} , with small k corresponding to a more flexible fit, and large k less flexible.

But it does have its limitations, an apparent one being that the fitted function \hat{m} essentially always looks jagged, especially for small or moderate k . Why is this? It helps to write

$$\hat{m}(x) = \sum_{i=1}^n w_i(x) Y_i, \quad (5)$$

where the weights $w_i(x)$, $i = 1, \dots, n$ are defined as

$$w_i(x) = \begin{cases} 1/k & \text{if } X_i \text{ is one of the } k \text{ nearest points to } x \\ 0 & \text{else.} \end{cases}$$

Note that $w_i(x)$ is discontinuous as a function of x , and therefore so is $\hat{m}(x)$.

The representation (5) also reveals that the k -nearest-neighbors estimate is in a class of estimates we call *linear smoothers*, i.e., writing $Y = (Y_1, \dots, Y_n) \in \mathbb{R}^n$, the vector of fitted values

$$\hat{\mu} = (\hat{m}(X_1), \dots, \hat{m}(X_n)) \in \mathbb{R}^n$$

can simply be expressed as $\hat{\mu} = SY$. (To be clear, this means that for fixed inputs X_1, \dots, X_n , the vector of fitted values $\hat{\mu}$ is a linear function of Y ; it does not mean that $\hat{m}(x)$ need behave linearly as a function of x .) This class is quite large, and contains many popular estimators, as we'll see in the coming sections.

The k -nearest-neighbors estimator is *universally consistent*, which means $\mathbb{E}\|\hat{m} - m_0\|_2^2 \rightarrow 0$ as $n \rightarrow \infty$, with no assumptions other than $\mathbb{E}(Y^2) \leq \infty$, provided that we take $k = k_n$ such that $k_n \rightarrow \infty$ and $k_n/n \rightarrow 0$; e.g., $k = \sqrt{n}$ will do. See Chapter 6.2 of [Gyorfi et al. \(2002\)](#).

Furthermore, assuming the underlying regression function m_0 is Lipschitz continuous, the k -nearest-neighbors estimate with $k \asymp n^{2/(2+d)}$ satisfies

$$\mathbb{E}\|\hat{m} - m_0\|_2^2 \lesssim n^{-2/(2+d)}. \quad (6)$$

See Chapter 6.3 of [Gyorfi et al. \(2002\)](#). Later, we will see that this is optimal.

Proof sketch: assume that $\text{Var}(Y|X = x) = \sigma^2$, a constant, for simplicity, and fix (condition on) the training points. Using the bias-variance tradeoff,

$$\begin{aligned} \mathbb{E}[(\hat{m}(x) - m_0(x))^2] &= \underbrace{(\mathbb{E}[\hat{m}(x)] - m_0(x))^2}_{\text{Bias}^2(\hat{m}(x))} + \underbrace{\mathbb{E}[(\hat{m}(x) - \mathbb{E}[\hat{m}(x)])^2]}_{\text{Var}(\hat{m}(x))} \\ &= \left(\frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} (m_0(X_i) - m_0(x)) \right)^2 + \frac{\sigma^2}{k} \\ &\leq \left(\frac{L}{k} \sum_{i \in \mathcal{N}_k(x)} \|X_i - x\|_2 \right)^2 + \frac{\sigma^2}{k}. \end{aligned}$$

In the last line we used the Lipschitz property $|m_0(x) - m_0(z)| \leq L\|x - z\|_2$, for some constant $L > 0$. Now for “most” of the points we'll have $\|X_i - x\|_2 \leq C(k/n)^{1/d}$, for a

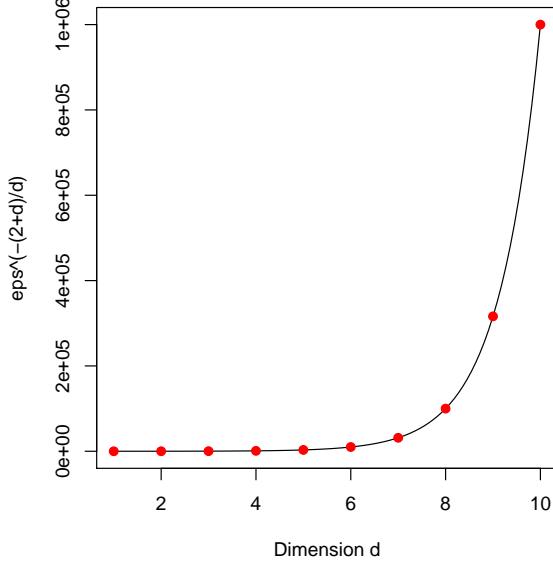


Figure 1: *The curse of dimensionality, with $\epsilon = 0.1$*

constant $C > 0$. (Think of having input points $X_i, i = 1, \dots, n$ spaced equally over (say) $[0, 1]^d$.) Then our bias-variance upper bound becomes

$$(CL)^2 \left(\frac{k}{n} \right)^{2/d} + \frac{\sigma^2}{k},$$

We can minimize this by balancing the two terms so that they are equal, giving $k^{1+2/d} \asymp n^{2/d}$, i.e., $k \asymp n^{2/(2+d)}$ as claimed. Plugging this in gives the error bound of $n^{-2/(2+d)}$, as claimed.

2.1 Curse of dimensionality

Note that the above error rate $n^{-2/(2+d)}$ exhibits a very poor dependence on the dimension d . To see it differently: given a small $\epsilon > 0$, think about how large we need to make n to ensure that $n^{-2/(2+d)} \leq \epsilon$. Rearranged, this says $n \geq \epsilon^{-(2+d)/2}$. That is, as we increase d , we require *exponentially more samples* n to achieve an error bound of ϵ . See Figure 1 for an illustration with $\epsilon = 0.1$.

In fact, this phenomenon is not specific to k -nearest-neighbors, but a reflection of the *curse of dimensionality*, the principle that estimation becomes exponentially harder as the number of dimensions increases. This is made precise by minimax theory: we cannot hope to do better than the rate in (6) over $H_d(1, L)$, which we write for the space of L -Lipschitz functions in d dimensions, for a constant $L > 0$. It can be shown that

$$\inf_{\hat{m}} \sup_{m_0 \in H_d(1, L)} \mathbb{E} \|\hat{m} - m_0\|_2^2 \gtrsim n^{-2/(2+d)}, \quad (7)$$

where the infimum above is over all estimators \hat{m} . See Chapter 3.2 of Gyorfi et al. (2002).

So why can we sometimes predict well in high dimensional problems? Presumably, it is because m_0 often (approximately) satisfies stronger assumptions. This suggests we should

look at classes of functions with more structure. One such example is the additive model, covered later in the notes.

3 Kernel Smoothing and Local Polynomials

3.1 Kernel smoothing

Kernel regression or *kernel smoothing* begins with a kernel function $K : \mathbb{R} \rightarrow \mathbb{R}$, satisfying

$$\int K(t) dt = 1, \quad \int tK(t) dt = 0, \quad 0 < \int t^2 K(t) dt < \infty.$$

Three common examples are the box-car kernel:

$$K(t) = \begin{cases} 1 & |x| \leq 1/2 \\ 0 & \text{otherwise} \end{cases},$$

the Gaussian kernel:

$$K(t) = \frac{1}{\sqrt{2\pi}} \exp(-t^2/2),$$

and the Epanechnikov kernel:

$$K(t) = \begin{cases} 3/4(1-t^2) & \text{if } |t| \leq 1 \\ 0 & \text{else} \end{cases}$$

Warning! Don't confuse this with the notion of kernels in RKHS methods which we cover later.

Given a bandwidth $h > 0$, the (Nadaraya-Watson) kernel regression estimate is defined as

$$\hat{m}(x) = \frac{\sum_{i=1}^n K\left(\frac{\|x - X_i\|_2}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{\|x - X_i\|_2}{h}\right)} = \sum_i w_i(x) Y_i \quad (8)$$

where $w_i(x) = K(\|x - X_i\|_2/h) / \sum_{j=1}^n K(\|x - X_j\|_2/h)$. Hence kernel smoothing is also a linear smoother.

In comparison to the k -nearest-neighbors estimator in (4), which can be thought of as a raw (discontinuous) moving average of nearby responses, the kernel estimator in (8) is a smooth moving average of responses. See Figure 2 for an example with $d = 1$.

3.2 Error Analysis

The kernel smoothing estimator is universally consistent ($\mathbb{E}\|\hat{m} - m_0\|_2^2 \rightarrow 0$ as $n \rightarrow \infty$, with no assumptions other than $\mathbb{E}(Y^2) \leq \infty$), provided we take a compactly supported kernel K , and bandwidth $h = h_n$ satisfying $h_n \rightarrow 0$ and $nh_n^d \rightarrow \infty$ as $n \rightarrow \infty$. See Chapter 5.2 of Gyorfi et al. (2002). We can say more.

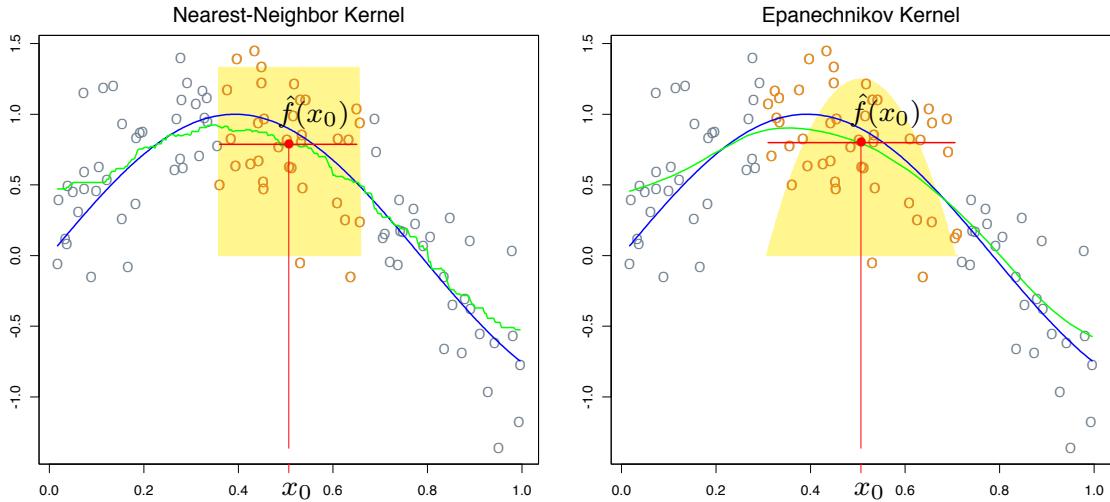


Figure 2: Comparing k -nearest-neighbor and Epanechnikov kernels, when $d = 1$. From Chapter 6 of [Hastie et al. \(2009\)](#)

Theorem. Suppose that $d = 1$ and that m'' is bounded. Also suppose that X has a non-zero, differentiable density p and that the support is unbounded. Then, the risk is

$$R_n = \frac{h_n^4}{4} \left(\int x^2 K(x) dx \right)^2 \int \left(m''(x) + 2m'(x) \frac{p'(x)}{p(x)} \right)^2 dx \\ + \frac{\sigma^2 \int K^2(x) dx}{nh_n} \int \frac{dx}{p(x)} + o\left(\frac{1}{nh_n}\right) + o(h_n^4)$$

where p is the density of P_X .

The first term is the squared bias. The dependence on p and p' is the design bias and is undesirable. We'll fix this problem later using local linear smoothing. It follows that the optimal bandwidth is $h_n \approx n^{-1/5}$ yielding a risk of $n^{-4/5}$. In d dimensions, the term nh_n becomes nh_n^d . In that case it follows that the optimal bandwidth is $h_n \approx n^{-1/(4+d)}$ yielding a risk of $n^{-4/(4+d)}$.

If the support has boundaries then there is bias of order $O(h)$ near the boundary. This happens because of the asymmetry of the kernel weights in such regions. See Figure 3. Specifically, the bias is of order $O(h^2)$ in the interior but is of order $O(h)$ near the boundaries. The risk then becomes $O(h^3)$ instead of $O(h^4)$. We'll fix this problems using local linear smoothing. Also, the result above depends on assuming that P_X has a density. We can drop that assumption (and allow for boundaries) and get a slightly weaker result due to Gyorfi, Kohler, Krzyzak and Walk (2002).

For simplicity, we will use the spherical kernel $K(\|x\|) = I(\|x\| \leq 1)$; the results can be extended to other kernels. Hence,

$$\hat{m}(x) = \frac{\sum_{i=1}^n Y_i I(\|X_i - x\| \leq h)}{\sum_{i=1}^n I(\|X_i - x\| \leq h)} = \frac{\sum_{i=1}^n Y_i I(\|X_i - x\| \leq h)}{n P_n(B(x, h))}$$

where P_n is the empirical measure and $B(x, h) = \{u : \|x - u\| \leq h\}$. If the denominator is 0 we define $\hat{m}(x) = 0$. The proof of the following theorem is from Chapter 5 of Györfi, Kohler, Krzyżak and Walk (2002).

Theorem: Risk bound without density. Suppose that the distribution of X has compact support and that $\text{Var}(Y|X = x) \leq \sigma^2 < \infty$ for all x . Then

$$\sup_{P \in H_d(1, L)} \mathbb{E} \|\hat{m} - m\|_P^2 \leq c_1 h^2 + \frac{c_2}{nh^d}. \quad (9)$$

Hence, if $h \asymp n^{-1/(d+2)}$ then

$$\sup_{P \in H_d(1, L)} \mathbb{E} \|\hat{m} - m\|_P^2 \leq \frac{c}{n^{2/(d+2)}}. \quad (10)$$

The proof is in the appendix. Note that the rate $n^{-2/(d+2)}$ is slower than the pointwise rate $n^{-4/(d+2)}$ because we have made weaker assumptions.

Recall from (7) we saw that this was the minimax optimal rate over $H_d(1, L)$. More generally, the minimax rate over $H_d(\alpha, L)$, for a constant $L > 0$, is

$$\inf_{\hat{m}} \sup_{m_0 \in H_d(\alpha, L)} \mathbb{E} \|\hat{m} - m_0\|_2^2 \gtrsim n^{-2\alpha/(2\alpha+d)}, \quad (11)$$

see again Chapter 3.2 of Gyorfi et al. (2002). However, as we saw above, with extra conditions, we got the rate $n^{-4/(4+d)}$ which is minimax for $H_d(2, L)$. We'll get that rate under weaker conditions with local linear regression.

If the support of the distribution of X lives on a smooth manifold of dimension $r < d$ then the term

$$\int \frac{dP(x)}{nP(B(x, h))}$$

is of order $1/(nh^r)$ instead of $1/(nh^d)$. In that case, we get the improved rate $n^{-2/(r+2)}$.

3.3 Local Linear Regression

We can alleviate this boundary bias issue by moving from a local constant fit to a local linear fit, or a local polynomial fit.

To build intuition, another way to view the kernel estimator in (8) is the following: at each input x , define the estimate $\hat{m}(x) = \hat{\theta}_x$, where $\hat{\theta}_x$ is the minimizer of

$$\sum_{i=1}^n K\left(\frac{\|x - X_i\|}{h}\right) (Y_i - \theta)^2,$$

over all $\theta \in \mathbb{R}$. In other words, Instead we could consider forming the local estimate $\hat{m}(x) = \hat{\alpha}_x + \hat{\beta}_x^T x$, where $\hat{\alpha}_x, \hat{\beta}_x$ minimize

$$\sum_{i=1}^n K\left(\frac{\|x - X_i\|}{h}\right) (Y_i - \alpha - \beta^T X_i)^2.$$

over all $\alpha \in \mathbb{R}, \beta \in \mathbb{R}^d$. This is called *local linear regression*.

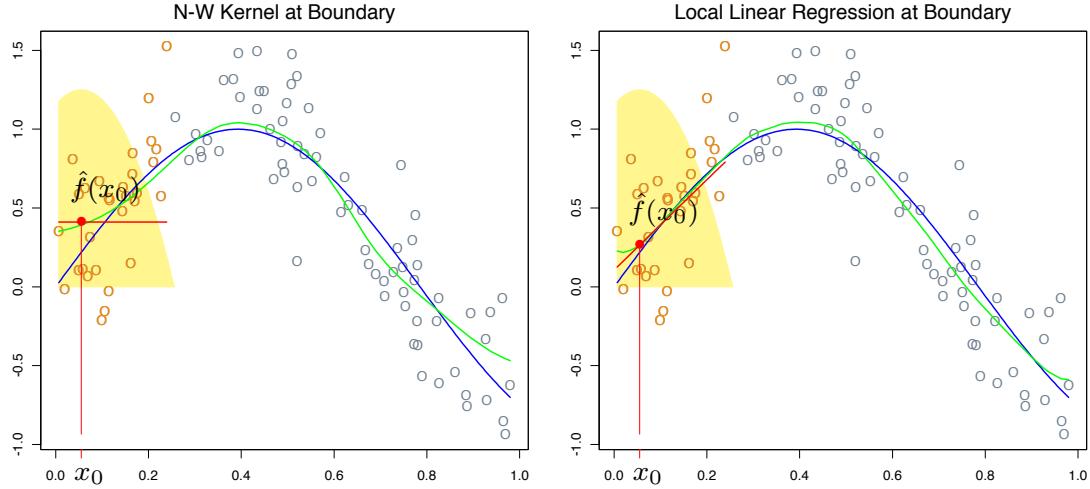


Figure 3: Comparing (Nadaraya-Watson) kernel smoothing to local linear regression; the former is biased at the boundary, the latter is unbiased (to first-order). From Chapter 6 of [Hastie et al. \(2009\)](#)

We can rewrite the local linear regression estimate $\hat{m}(x)$. This is just given by a weighted least squares fit, so

$$\hat{m}(x) = b(x)^T (B^T \Omega B)^{-1} B^T \Omega Y,$$

where $b(x) = (1, x) \in \mathbb{R}^{d+1}$, $B \in \mathbb{R}^{n \times (d+1)}$ with i th row $b(X_i)$, and $\Omega \in \mathbb{R}^{n \times n}$ is diagonal with i th diagonal element $K(\|x - X_i\|_2/h)$. We can write more concisely as $\hat{m}(x) = w(x)^T Y$, where $w(x) = \Omega B(B^T \Omega B)^{-1} b(x)$, which shows local linear regression is a linear smoother too.

The vector of fitted values $\hat{\mu} = (\hat{m}(x_1), \dots, \hat{m}(x_n))$ can be expressed as

$$\hat{\mu} = \begin{pmatrix} w_1(x)^T Y \\ \vdots \\ w_n(x)^T Y \end{pmatrix} = B(B^T \Omega B)^{-1} B^T \Omega Y = SY$$

which should look familiar to you from weighted least squares.

Now we'll sketch how the local linear fit reduces the bias, fixing (conditioning on) the training points. Compute at a fixed point x ,

$$\mathbb{E}[\hat{m}(x)] = \sum_{i=1}^n w_i(x) m_0(X_i).$$

Using a Taylor expansion of m_0 about x ,

$$\mathbb{E}[\hat{m}(x)] = m_0(x) \sum_{i=1}^n w_i(x) + \nabla m_0(x)^T \sum_{i=1}^n (X_i - x) w_i(x) + R,$$

where the remainder term R contains quadratic and higher-order terms, and under regularity conditions, is small. One can check that in fact for the local linear regression estimator \hat{m} ,

$$\sum_{i=1}^n w_i(x) = 1 \quad \text{and} \quad \sum_{i=1}^n (X_i - x) w_i(x) = 0,$$

and so $\mathbb{E}[\hat{m}(x)] = m_0(x) + R$, which means that \hat{m} is unbiased to first-order.

It can be shown that local linear regression removes boundary bias and design bias.

Theorem. Under some regularity conditions, the risk of \hat{m} is

$$\frac{h_n^4}{4} \int \left(\text{tr}(m''(x) \int K(u) u u^T du) \right)^2 dP(x) + \frac{1}{nh_n^d} \int K^2(u) du \int \sigma^2(x) dP(x) + o(h_n^4 + (nh_n^d)^{-1}).$$

For a proof, see [Fan & Gijbels \(1996\)](#). For points near the boundary, the bias is $Ch^2 m''(x) + o(h^2)$ whereas, the bias is $Chm'(x) + o(h)$ for kernel estimators.

In fact, [Fan \(1993\)](#) shows a rather remarkable result. Let R_n be the minimax risk for estimating $m(x_0)$ over the class of functions with bounded second derivatives in a neighborhood of x_0 . Let the maximum risk r_n of the local linear estimator with optimal bandwidth satisfies

$$1 + o(1) \geq \frac{R_n}{r_n} \geq (0.896)^2 + o(1).$$

Moreover, if we compute the minimax risk over all linear estimators we get $\frac{R_n}{r_n} \rightarrow 1$.

3.4 Higher-order smoothness

How can we hope to get optimal error rates over $H_d(\alpha, d)$, when $\alpha \geq 2$? With kernels there are basically two options: use local polynomials, or use higher-order kernels

Local polynomials build on our previous idea of local linear regression (itself an extension of kernel smoothing.) Consider $d = 1$, for concreteness. Define $\hat{m}(x) = \hat{\beta}_{x,0} + \sum_{j=1}^k \hat{\beta}_{x,j} x^j$, where $\hat{\beta}_{x,0}, \dots, \hat{\beta}_{x,k}$ minimize

$$\sum_{i=1}^n K\left(\frac{|x - X_i|}{h}\right) \left(Y_i - \beta_0 - \sum_{j=1}^k \beta_j X_i^j\right)^2.$$

over all $\beta_0, \beta_1, \dots, \beta_k \in \mathbb{R}$. This is called (k th-order) *local polynomial regression*

Again we can express

$$\hat{m}(x) = b(x)(B^T \Omega B)^{-1} B^T \Omega y = w(x)^T y,$$

where $b(x) = (1, x, \dots, x^k)$, B is an $n \times (k+1)$ matrix with i th row $b(X_i) = (1, X_i, \dots, X_i^k)$, and Ω is as before. Hence again, local polynomial regression is a linear smoother

Assuming that $m_0 \in H_1(\alpha, L)$ for a constant $L > 0$, a Taylor expansion shows that the local polynomial estimator \hat{m} of order k , where k is the largest integer strictly less than α and where the bandwidth scales as $h \asymp n^{-1/(2\alpha+1)}$, satisfies

$$\mathbb{E} \|\hat{m} - m_0\|_2^2 \lesssim n^{-2\alpha/(2\alpha+1)}.$$

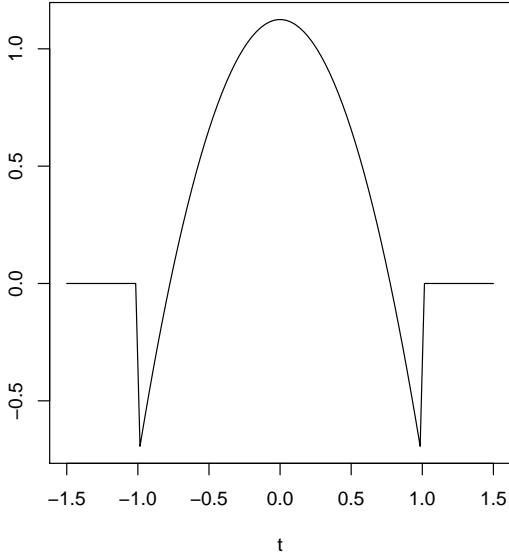


Figure 4: A higher-order kernel function: specifically, a kernel of order 4

See Chapter 1.6.1 of [Tsybakov \(2009\)](#). This matches the lower bound in (11) (when $d = 1$)

In multiple dimensions, $d > 1$, local polynomials become kind of tricky to fit, because of the explosion in terms of the number of parameters we need to represent a k th order polynomial in d variables. Hence, an interesting alternative is to return back kernel smoothing but use a *higher-order kernel*. A kernel function K is said to be of order k provided that

$$\int K(t) dt = 1, \quad \int t^j K(t) dt = 0, \quad j = 1, \dots, k-1, \quad \text{and} \quad 0 < \int t^k K(t) dt < \infty.$$

This means that the kernels we were looking at so far were of order 2

An example of a 4th-order kernel is $K(t) = \frac{3}{8}(3 - 5t^2)\mathbf{1}\{|t| \leq 1\}$, plotted in Figure 4. Notice that it takes negative values.

Lastly, while local polynomial regression and higher-order kernel smoothing can help “track” the derivatives of smooth functions $m_0 \in H_d(\alpha, L)$, $\alpha \geq 2$, it should be noted that they don’t share the same universal consistency property of kernel smoothing (or k -nearest-neighbors). See Chapters 5.3 and 5.4 of [Gyorfi et al. \(2002\)](#)

4 Splines

Suppose that $d = 1$. Define an estimator by

$$\hat{m} = \operatorname{argmin}_f \sum_{i=1}^n (Y_i - m(X_i))^2 + \lambda \int_0^1 m''(x)^2 dx. \quad (12)$$

Spline Lemma. The minimizer of (25) is a cubic spline with knots at the data points. (Proof in the Appendix.)

The key result presented above tells us that we can choose a basis η_1, \dots, η_n for the set of k th-order natural splines with knots over x_1, \dots, x_n , and reparametrize the problem as

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^n \left(Y_i - \sum_{j=1}^n \beta_j \eta_j(X_i) \right)^2 + \lambda \int_0^1 \left(\sum_{j=1}^n \beta_j \eta_j''(x) \right)^2 dx. \quad (13)$$

This is a finite-dimensional problem, and after we compute the coefficients $\hat{\beta} \in \mathbb{R}^n$, we know that the smoothing spline estimate is simply $\hat{m}(x) = \sum_{j=1}^n \hat{\beta}_j \eta_j(x)$

Defining the basis matrix and penalty matrices $N, \Omega \in \mathbb{R}^{n \times n}$ by

$$N_{ij} = \eta_j(X_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 \eta_i''(x) \eta_j''(x) dx \quad \text{for } i, j = 1, \dots, n, \quad (14)$$

the problem in (27) can be written more succinctly as

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \|Y - N\beta\|_2^2 + \lambda \beta \Omega \beta, \quad (15)$$

showing the smoothing spline problem to be a type of generalized ridge regression problem. In fact, the solution in (29) has the explicit form

$$\hat{\beta} = (N^T N + \lambda \Omega)^{-1} N^T Y,$$

and therefore the fitted values $\hat{\mu} = (\hat{m}(x_1), \dots, \hat{m}(x_n))$ are

$$\hat{\mu} = N(N^T N + \lambda \Omega)^{-1} N^T Y \equiv SY. \quad (16)$$

Therefore, once again, smoothing splines are a type of linear smoother

A special property of smoothing splines: the fitted values in (30) can be computed in $O(n)$ operations. This is achieved by forming N from the B-spline basis (for natural splines), and in this case the matrix $N^T N + \Omega I$ ends up being banded (with a bandwidth that only depends on the polynomial order k). In practice, smoothing spline computations are extremely fast

4.1 Error rates

Recall the *Sobolev class* of functions $S_1(m, C)$: for an integer $m \geq 0$ and $C > 0$, to contain all m times differentiable functions $f : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$\int (f^{(m)}(x))^2 dx \leq C^2.$$

(The Sobolev class $S_d(m, C)$ in d dimensions can be defined similarly, where we sum over all partial derivatives of order m .)

Assuming $m_0 \in S_1(m, C)$ for the underlying regression function, where $C > 0$ is a constant, the smoothing spline estimator \hat{m} of polynomial order $k = 2m - 1$ with tuning parameter $\lambda \asymp n^{1/(2m+1)} \asymp n^{1/(k+2)}$ satisfies

$$\|\hat{m} - m_0\|_n^2 \lesssim n^{-2m/(2m+1)} \quad \text{in probability.}$$

The proof of this result uses much more fancy techniques from empirical process theory (entropy numbers) than the proofs for kernel smoothing. See Chapter 10.1 of [van de Geer \(2000\)](#). This rate is seen to be minimax optimal over $S_1(m, C)$ (e.g., [Nussbaum \(1985\)](#)).

5 Mercer kernels, RKHS

5.1 Hilbert Spaces

A Hilbert space is a complete inner product space. We will see that a reproducing kernel Hilbert space (RKHS) is a Hilbert space with extra structure that makes it very useful for statistics and machine learning.

An example of a Hilbert space is

$$L_2[0, 1] = \left\{ f : [0, 1] \rightarrow \mathbb{R} : \int f^2 < \infty \right\}$$

endowed with the inner product

$$\langle f, g \rangle = \int f(x)g(x)dx.$$

The corresponding norm is

$$\|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int f^2(x)dx}.$$

We write $f_n \rightarrow f$ to mean that $\|f_n - f\| \rightarrow 0$ as $n \rightarrow \infty$.

5.2 Evaluation Functional

The evaluation functional δ_x assigns a real number to each function. It is defined by $\delta_x f = f(x)$. In general, the evaluation functional is not continuous. This means we can have $f_n \rightarrow f$ but $\delta_x f_n$ does not converge to $\delta_x f$. For example, let $f(x) = 0$ and $f_n(x) = \sqrt{n}I(x < 1/n^2)$. Then $\|f_n - f\| = 1/\sqrt{n} \rightarrow 0$. But $\delta_0 f_n = \sqrt{n}$ which does not converge to $\delta_0 f = 0$. Intuitively, this is because Hilbert spaces can contain very unsophisticated functions. We shall see that RKHS are Hilbert spaces where the evaluation functional is continuous. Intuitively, this means that the functions in the space are well-behaved.

5.3 Nonparametric Regression

We observe $(X_1, Y_1), \dots, (X_n, Y_n)$ and we want to estimate $m(x) = \mathbb{E}(Y|X = x)$. The approach we used earlier was based on **smoothing kernels**:

$$\hat{m}(x) = \frac{\sum_i Y_i K\left(\frac{\|x-X_i\|}{h}\right)}{\sum_i K\left(\frac{\|x-X_i\|}{h}\right)}.$$

Another approach is regularization: choose m to minimize

$$\sum_i (Y_i - m(X_i))^2 + \lambda J(m)$$

for some penalty J . This is equivalent to: choose $m \in \mathcal{M}$ to minimize $\sum_i (Y_i - m(X_i))^2$ where $\mathcal{M} = \{m : J(m) \leq L\}$ for some $L > 0$.

We would like to construct \mathcal{M} so that it contains smooth functions. We shall see that a good choice is to use a RKHS.

5.4 Mercer Kernels

A RKHS is defined by a **Mercer kernel**. A Mercer kernel $K(x, y)$ is a function of two variables that is symmetric and positive definite. This means that, for any function f ,

$$\int \int K(x, y) f(x) f(y) dx dy \geq 0.$$

(This is like the definition of a positive definite matrix: $x^T A x \geq 0$ for each x .)

Our main example is the Gaussian kernel

$$K(x, y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}.$$

Given a kernel K , let $K_x(\cdot)$ be the function obtained by fixing the first coordinate. That is, $K_x(y) = K(x, y)$. For the Gaussian kernel, K_x is a Normal, centered at x . We can create functions by taking linear combinations of the kernel:

$$f(x) = \sum_{j=1}^k \alpha_j K_{x_j}(x).$$

Let \mathcal{H}_0 denote all such functions:

$$\mathcal{H}_0 = \left\{ f : \sum_{j=1}^k \alpha_j K_{x_j}(x) \right\}.$$

Given two such functions $f(x) = \sum_{j=1}^k \alpha_j K_{x_j}(x)$ and $g(x) = \sum_{j=1}^m \beta_j K_{y_j}(x)$ we define an inner product

$$\langle f, g \rangle = \langle f, g \rangle_K = \sum_i \sum_j \alpha_i \beta_j K(x_i, y_j).$$

In general, f (and g) might be representable in more than one way. You can check that $\langle f, g \rangle_K$ is independent of how f (or g) is represented. The inner product defines a norm:

$$\|f\|_K = \sqrt{\langle f, f \rangle} = \sqrt{\sum_j \sum_k \alpha_j \alpha_k K(x_j, x_k)} = \sqrt{\alpha^T \mathbb{K} \alpha}$$

where $\alpha = (\alpha_1, \dots, \alpha_k)^T$ and \mathbb{K} is the $k \times k$ matrix with $\mathbb{K}_{jk} = K(x_j, x_k)$.

5.5 The Reproducing Property

Let $f(x) = \sum_i \alpha_i K_{x_i}(x)$. Note the following crucial property:

$$\langle f, K_x \rangle = \sum_i \alpha_i K(x_i, x) = f(x).$$

This follows from the definition of $\langle f, g \rangle$ where we take $g = K_x$. This implies that

$$\langle K_x, K_y \rangle = K(x, y).$$

This is called the reproducing property. It also implies that K_x is the **representer** of the evaluation functional.

The completion of \mathcal{H}_0 with respect to $\|\cdot\|_K$ is denoted by \mathcal{H}_K and is called the RKHS generated by K .

To verify that this is a well-defined Hilbert space, you should check that the following properties hold:

$$\begin{aligned}\langle f, g \rangle &= \langle g, f \rangle \\ \langle cf + dg, h \rangle &= c\langle f, h \rangle + d\langle g, h \rangle \\ \langle f, f \rangle &= 0 \text{ iff } f = 0.\end{aligned}$$

The last one is not obvious so let us verify it here. It is easy to see that $f = 0$ implies that $\langle f, f \rangle = 0$. Now we must show that $\langle f, f \rangle = 0$ implies that $f(x) = 0$. So suppose that $\langle f, f \rangle = 0$. Pick any x . Then

$$\begin{aligned}0 &\leq f^2(x) = \langle f, K_x \rangle^2 = \langle f, K_x \rangle \langle f, K_x \rangle \\ &\leq \|f\|^2 \|K_x\|^2 = \langle f, f \rangle^2 \|K_x\|^2 = 0\end{aligned}$$

where we used Cauchy-Schwartz. So $0 \leq f^2(x) \leq 0$ which means that $f(x) = 0$.

Returning to the evaluation functional, suppose that $f_n \rightarrow f$. Then

$$\delta_x f_n = \langle f_n, K_x \rangle \rightarrow \langle f, K_x \rangle = f(x) = \delta_x f$$

so the evaluation functional is continuous. **In fact, a Hilbert space is a RKHS if and only if the evaluation functionals are continuous.**

5.6 Examples

Example 1. Let \mathcal{H} be all functions f on \mathbb{R} such that the support of the Fourier transform of f is contained in $[-a, a]$. Then

$$K(x, y) = \frac{\sin(a(y - x))}{a(y - x)}$$

and

$$\langle f, g \rangle = \int fg.$$

Example 2. Let \mathcal{H} be all functions f on $(0, 1)$ such that

$$\int_0^1 (f^2(x) + (f'(x))^2)x^2 dx < \infty.$$

Then

$$K(x, y) = (xy)^{-1} (e^{-x} \sinh(y) I(0 < x \leq y) + e^{-y} \sinh(x) I(0 < y \leq x))$$

and

$$\|f\|^2 = \int_0^1 (f^2(x) + (f'(x))^2)x^2 dx.$$

Example 3. The Sobolev space of order m is (roughly speaking) the set of functions f such that $\int(f^{(m)})^2 < \infty$. For $m = 1$ and $\mathcal{X} = [0, 1]$ the kernel is

$$K(x, y) = \begin{cases} 1 + xy + \frac{xy^2}{2} - \frac{y^3}{6} & 0 \leq y \leq x \leq 1 \\ 1 + xy + \frac{yx^2}{2} - \frac{x^3}{6} & 0 \leq x \leq y \leq 1 \end{cases}$$

and

$$\|f\|_K^2 = f^2(0) + f'(0)^2 + \int_0^1 (f''(x))^2 dx.$$

5.7 Spectral Representation

Suppose that $\sup_{x,y} K(x, y) < \infty$. Define eigenvalues λ_j and orthonormal eigenfunctions ψ_j by

$$\int K(x, y)\psi_j(y)dy = \lambda_j\psi_j(x).$$

Then $\sum_j \lambda_j < \infty$ and $\sup_x |\psi_j(x)| < \infty$. Also,

$$K(x, y) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x) \psi_j(y).$$

Define the **feature map** Φ by

$$\Phi(x) = (\sqrt{\lambda_1}\psi_1(x), \sqrt{\lambda_2}\psi_2(x), \dots).$$

We can expand f either in terms of K or in terms of the basis ψ_1, ψ_2, \dots :

$$f(x) = \sum_i \alpha_i K(x_i, x) = \sum_{j=1}^{\infty} \beta_j \psi_j(x).$$

Furthermore, if $f(x) = \sum_j a_j \psi_j(x)$ and $g(x) = \sum_j b_j \psi_j(x)$, then

$$\langle f, g \rangle = \sum_{j=1}^{\infty} \frac{a_j b_j}{\lambda_j}.$$

Roughly speaking, when $\|f\|_K$ is small, then f is smooth.

5.8 Representer Theorem

Let ℓ be a loss function depending on $(X_1, Y_1), \dots, (X_n, Y_n)$ and on $f(X_1), \dots, f(X_n)$. Let \hat{f} minimize

$$\ell + g(\|f\|_K^2)$$

where g is any monotone increasing function. Then \hat{f} has the form

$$\hat{f}(x) = \sum_{i=1}^n \alpha_i K(x_i, x)$$

for some $\alpha_1, \dots, \alpha_n$.

5.9 RKHS Regression

Define \hat{m} to minimize

$$R = \sum_i (Y_i - m(X_i))^2 + \lambda \|m\|_K^2.$$

By the representer theorem, $\hat{m}(x) = \sum_{i=1}^n \alpha_i K(x_i, x)$. Plug this into R and we get

$$R = \|Y - \mathbb{K}\alpha\|^2 + \lambda \alpha^T \mathbb{K} \alpha$$

where $\mathbb{K}_{jk} = K(X_j, X_k)$ is the Gram matrix. The minimizer over α is

$$\hat{\alpha} = (\mathbb{K} + \lambda I)^{-1} Y$$

and $\hat{m}(x) = \sum_j \hat{\alpha}_j K(X_i, x)$. The fitted values are

$$\hat{Y} = \mathbb{K}\hat{\alpha} = \mathbb{K}(\mathbb{K} + \lambda I)^{-1} Y = LY.$$

So this is a linear smoother.

We can use cross-validation to choose λ . **Compare this with smoothing kernel regression.**

5.10 Logistic Regression

Let

$$m(x) = \mathbb{P}(Y = 1 | X = x) = \frac{e^{f(x)}}{1 + e^{f(x)}}.$$

We can estimate m by minimizing

$$-\text{loglikelihood} + \lambda \|f\|_K^2.$$

Then $\hat{f} = \sum_j K(x_j, x)$ and α may be found by numerical optimization. In this case, smoothing kernels are much easier.

5.11 Support Vector Machines

Suppose $Y_i \in \{-1, +1\}$. Recall the the linear SVM minimizes the penalized hinge loss:

$$J = \sum_i [1 - Y_i(\beta_0 + \beta^T X_i)]_+ + \frac{\lambda}{2} \|\beta\|_2^2.$$

The dual is to maximize

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j \langle X_i, X_j \rangle$$

subject to $0 \leq \alpha_i \leq C$.

The RKHS version is to minimize

$$J = \sum_i [1 - Y_i f(X_i)]_+ + \frac{\lambda}{2} \|f\|_K^2.$$

The dual is the same except that $\langle X_i, X_j \rangle$ is replaced with $K(X_i, X_j)$. This is called the kernel trick.

5.12 The Kernel Trick

This is a fairly general trick. In many algorithms you can replace $\langle x_i, x_j \rangle$ with $K(x_i, x_j)$ and get a nonlinear version of the algorithm. This is equivalent to replacing x with $\Phi(x)$ and replacing $\langle x_i, x_j \rangle$ with $\langle \Phi(x_i), \Phi(x_j) \rangle$. However, $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ and $K(x_i, x_j)$ is much easier to compute.

In summary, by replacing $\langle x_i, x_j \rangle$ with $K(x_i, x_j)$ we turn a linear procedure into a nonlinear procedure without adding much computation.

5.13 Hidden Tuning Parameters

There are hidden tuning parameters in the RKHS. Consider the Gaussian kernel

$$K(x, y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}.$$

For nonparametric regression we minimize $\sum_i (Y_i - m(X_i))^2$ subject to $\|m\|_K \leq L$. We control the bias variance tradeoff by doing cross-validation over L . But what about σ ?

This parameter seems to get mostly ignored. Suppose we have a uniform distribution on a circle. The eigenfunctions of $K(x, y)$ are the sines and cosines. The eigenvalues λ_k die off like $(1/\sigma)^{2k}$. So σ affects the bias-variance tradeoff since it weights things towards lower order Fourier functions. In principle we can compensate for this by varying L . But clearly there is some interaction between L and σ . The practical effect is not well understood. We'll see this again when we discuss interpolation.

Now consider the polynomial kernel $K(x, y) = (1 + \langle x, y \rangle)^d$. This kernel has the same eigenfunctions but the eigenvalues decay at a polynomial rate depending on d . So there is an interaction between L , d and, the choice of kernel itself.

6 Linear smoothers

6.1 Definition

Every estimator we have discussed so far is a linear smoother meaning that $\hat{m}(x) = \sum_i w_i(x)Y_i$ for some weights $w_i(x)$ that do not depend on the Y_i 's. Hence, the fitted values $\hat{\mu} = (\hat{m}(X_1), \dots, \hat{m}(X_n))$ are of the form $\hat{\mu} = SY$ for some matrix $S \in \mathbb{R}^{n \times n}$ depending on the inputs X_1, \dots, X_n —and also possibly on a tuning parameter such as h in kernel smoothing, or λ in smoothing splines—but not on the Y_i 's. We call S , the smoothing matrix. For comparison, recall that in linear regression, $\hat{\mu} = HY$ for some projection matrix H .

For linear smoothers $\hat{\mu} = SY$, the effective degrees of freedom is defined to be

$$\nu \equiv \text{df}(\hat{\mu}) \equiv \sum_{i=1}^n S_{ii} = \text{tr}(S),$$

the trace of the smooth matrix S

6.2 Cross-validation

K -fold cross-validation can be used to estimate the prediction error and choose tuning parameters.

For linear smoothers $\hat{\mu} = (\hat{m}(x_1), \dots, \hat{m}(x_n)) = SY$, leave-one-out cross-validation can be particularly appealing because in many cases we have the seemingly magical reduction

$$\text{CV}(\hat{m}) = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{m}^{-i}(X_i))^2 = \frac{1}{n} \sum_{i=1}^n \left(\frac{Y_i - \hat{m}(X_i)}{1 - S_{ii}} \right)^2, \quad (17)$$

where \hat{m}^{-i} denotes the estimated regression function that was trained on all but the i th pair (X_i, Y_i) . This leads to a big computational savings since it shows us that, to compute leave-one-out cross-validation error, we don't have to actually ever compute \hat{m}^{-i} , $i = 1, \dots, n$.

Why does (17) hold, and for which linear smoothers $\hat{\mu} = Sy$? Just rearranging (17) perhaps demystifies this seemingly magical relationship and helps to answer these questions. Suppose we knew that \hat{m} had the property

$$\hat{m}^{-i}(X_i) = \frac{1}{1 - S_{ii}} (\hat{m}(X_i) - S_{ii}Y_i). \quad (18)$$

That is, to obtain the estimate at X_i under the function \hat{m}^{-i} fit on all but (X_i, Y_i) , we take the sum of the linear weights (from our original fitted function \hat{m}) across all but the i th point, $\hat{m}(X_i) - S_{ii}Y_i = \sum_{j \neq i} S_{ij}Y_j$, and then renormalize so that these weights sum to 1.

This is not an unreasonable property; e.g., we can immediately convince ourselves that it holds for kernel smoothing. A little calculation shows that it also holds for smoothing splines (using the Sherman-Morrison update formula).

From the special property (18), it is easy to show the leave-one-out formula (17). We have

$$Y_i - \hat{m}^{-i}(X_i) = Y_i - \frac{1}{1 - S_{ii}} (\hat{m}(X_i) - S_{ii}Y_i) = \frac{Y_i - \hat{m}(X_i)}{1 - S_{ii}},$$

and then squaring both sides and summing over n gives (17).

Finally, *generalized cross-validation* is a small twist on the right-hand side in (17) that gives an approximation to leave-one-out cross-validation error. It is defined as by replacing the appearances of diagonal terms S_{ii} with the average diagonal term $\text{tr}(S)/n$,

$$\text{GCV}(\hat{m}) = \frac{1}{n} \sum_{i=1}^n \left(\frac{Y_i - \hat{m}(X_i)}{1 - \text{tr}(S)/n} \right)^2 = (1 - \nu/n)^{-2} \hat{R}$$

where ν is the effective degrees of freedom and \hat{R} is the training error. This can be of computational advantage in some cases where $\text{tr}(S)$ is easier to compute than individual elements S_{ii} .

7 Additive models

7.1 Motivation and definition

Computational efficiency and statistical efficiency are both very real concerns as the dimension d grows large, in nonparametric regression. If you're trying to fit a kernel, thin-plate

spline, or RKHS estimate in > 20 dimensions, without any other kind of structural constraints, then you'll probably be in trouble (unless you have a very fast computer and tons of data).

Recall from (11) that the minimax rate over the Holder class $H_d(\alpha, L)$ is $n^{-2\alpha/(2\alpha+d)}$, which has an exponentially bad dependence on the dimension d . This is usually called the curse of dimensionality (though the term apparently originated with [Bellman \(1962\)](#), who encountered an analogous issue but in a separate context—dynamic programming).

What can we do? One answer is to change what we're looking for, and fit estimates with less flexibility in high dimensions. Think of a linear model in d variables: there is a big difference between this and a fully nonparametric model in d variables. Is there some middle man that we can consider, that would make sense?

Additive models play the role of this middle man. Instead of considering a full d -dimensional function of the form

$$m(x) = m(x(1), \dots, x(d)) \quad (19)$$

we restrict our attention to functions of the form

$$m(x) = m_1(x(1)) + \dots + m_d(x(d)). \quad (20)$$

As each function m_j , $j = 1, \dots, d$ is univariate, fitting an estimate of the form (20) is certainly less ambitious than fitting one of the form (19). On the other hand, the scope of (20) is still big enough that we can capture interesting (marginal) behavior in high dimensions.

There is a link to naive-Bayes classification that we will discuss later.

The choice of estimator of the form (20) need not be regarded as an assumption we make about the true function m_0 , just like we don't always assume that the true model is linear when using linear regression. In many cases, we fit an additive model because we think it may provide a useful approximation to the truth, and is able to scale well with the number of dimensions d .

A classic result by [Stone \(1985\)](#) encapsulates this idea precisely. He shows that, while it may be difficult to estimate an arbitrary regression function m_0 in multiple dimensions, we can still estimate its *best additive approximation* \bar{m}^{add} well. Assuming each component function $\bar{m}_{0,j}^{\text{add}}$, $j = 1, \dots, d$ lies in the Holder class $H_1(\alpha, L)$, for constant $L > 0$, and we can use an additive model, with each component \hat{m}_j , $j = 1, \dots, d$ estimated using an appropriate k th degree spline, to give

$$\mathbb{E}\|\hat{m}_j - \bar{m}_j^{\text{add}}\|_2^2 \lesssim n^{-2\alpha/(2\alpha+1)}, \quad j = 1, \dots, d.$$

Hence each component of the best additive approximation \bar{f}^{add} to m_0 can be estimated at the optimal univariate rate. Loosely speaking, though we cannot hope to recover m_0 arbitrarily, we can recover its major structure along the coordinate axes.

7.2 Backfitting

Estimation with additive models is actually very simple; we can just choose our favorite univariate smoother (i.e., nonparametric estimator), and cycle through estimating each

function m_j , $j = 1, \dots, d$ individually (like a block coordinate descent algorithm). Denote the result of running our chosen univariate smoother to regress $Y = (Y_1, \dots, Y_n) \in \mathbb{R}^n$ over the input points $Z = (Z_1, \dots, Z_n) \in \mathbb{R}^n$ as

$$\hat{m} = \text{Smooth}(Z, Y).$$

E.g., we might choose $\text{Smooth}(\cdot, \cdot)$ to be a cubic smoothing spline with some fixed value of the tuning parameter λ , or even with the tuning parameter selected by generalized cross-validation

Once our univariate smoother has been chosen, we initialize $\hat{m}_1, \dots, \hat{m}_d$ (say, to all to zero) and cycle over the following steps for $j = 1, \dots, d, 1, \dots, d, \dots$:

1. define $r_i = Y_i - \sum_{\ell \neq j} \hat{m}_\ell(x_{i\ell})$, $i = 1, \dots, n$;
2. smooth $\hat{m}_j = \text{Smooth}(x(j), r)$;
3. center $\hat{m}_j = \hat{m}_j - \frac{1}{n} \sum_{i=1}^n \hat{m}_j(X_i(j))$.

This algorithm is known as *backfitting*. In last step above, we are removing the mean from each fitted function \hat{m}_j , $j = 1, \dots, d$, otherwise the model would not be identifiable. Our final estimate therefore takes the form

$$\hat{m}(x) = \bar{Y} + \hat{m}_1(x(1)) + \dots + \hat{m}_d(x(d))$$

where $\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$. [Hastie & Tibshirani \(1990\)](#) provide a very nice exposition on the some of the more practical aspects of backfitting and additive models.

In many cases, backfitting is equivalent to blockwise coordinate descent performed on a joint optimization criterion that determines the total additive estimate. E.g., for the additive cubic smoothing spline optimization problem,

$$\hat{m}_1, \dots, \hat{m}_d = \underset{m_1, \dots, m_d}{\operatorname{argmin}} \sum_{i=1}^n \left(Y_i - \sum_{j=1}^d m_j(x_{ij}) \right)^2 + \sum_{j=1}^d \lambda_j \int_0^1 m_j''(t)^2 dt,$$

backfitting is exactly blockwise coordinate descent (after we reparametrize the above to be in finite-dimensional form, using a natural cubic spline basis).

The beauty of backfitting is that it allows us to think *algorithmically*, and plug in whatever we want for the univariate smoothers. This allows for several extensions. One extension: we don't need to use the same univariate smoother for each dimension, rather, we could mix and match, choosing $\text{Smooth}_j(\cdot, \cdot)$, $j = 1, \dots, d$ to come from entirely different methods or giving estimates with entirely different structures.

Another extension: to capture interactions, we can perform smoothing over (small) groups of variables instead of individual variables. For example we could fit a model of the form

$$m(x) = \sum_j m_j(x(j)) + \sum_{j < k} m_{jk}(x(j), x(k)).$$

7.3 Error rates

Error rates for additive models are both kind of what you'd expect and surprising. What you'd expect: if the underlying function m_0 is additive, and we place standard assumptions on its component functions, such as $f_{0,j} \in S_1(m, C)$, $j = 1, \dots, d$, for a constant $C > 0$, a somewhat straightforward argument building on univariate minimax theory gives us the lower bound

$$\inf_{\hat{m}} \sup_{m_0 \in \bigoplus_{j=1}^d S_1(m, C)} \mathbb{E} \|\hat{m} - m_0\|_2^2 \gtrsim dn^{-2m/(2m+1)}.$$

This is simply d times the univariate minimax rate. (Note that we have been careful to track the role of d here, i.e., it is not being treated like a constant.) Also, standard methods like backfitting with univariate smoothing splines of polynomial order $k = 2m - 1$, will also match this upper bound in error rate (though the proof to get the sharp linear dependence on d is a bit trickier).

7.4 Sparse additive models

Recently, *sparse additive models* have received a good deal of attention. In truly high dimensions, we might believe that only a small subset of the variables play a useful role in modeling the regression function, so might posit a modification of (20) of the form

$$m(x) = \sum_{j \in S} m_j(x(j))$$

where $S \subseteq \{1, \dots, d\}$ is an unknown subset of the full set of dimensions.

This is a natural idea, and to estimate a sparse additive model, we can use methods that are like nonparametric analogies of the lasso (more accurately, the group lasso). This is a research topic still very much in development; some recent works are [Lin & Zhang \(2006\)](#), [Ravikumar et al. \(2009\)](#), [Raskutti et al. \(2012\)](#). We'll cover this in more detail when we talk about the sparsity, the lasso, and high-dimensional estimation.

8 Variance Estimation and Confidence Bands

Let

$$\sigma^2(x) = \text{Var}(Y|X = x).$$

We can estimate $\sigma^2(x)$ as follows. Let $\hat{m}(x)$ be an estimate of the regression function. Let $e_i = Y_i - \hat{m}(X_i)$. Now apply nonparametric regression again treating e_i^2 as the response. The resulting estimator $\hat{\sigma}^2(x)$ can be shown to be consistent under some regularity conditions.

Ideally we would also like to find random functions ℓ_n and u_n such that

$$P(\ell_n(x) \leq m(x) \leq u_n(x) \text{ for all } x) \rightarrow 1 - \alpha.$$

For the reasons we discussed earlier with density functions, this is essentially an impossible problem.

We can, however, still get an informal (but useful) estimate the variability of $\hat{m}(x)$. Suppose that $\hat{m}(x) = \sum_i w_i(x)Y_i$. The conditional variance is $\sum_i w_i^2(x)\sigma^2(x)$ which can

be estimated by $\sum_i w_i^2(x) \hat{\sigma}^2(x)$. An asymptotic, pointwise (biased) confidence band is $\hat{m}(x) \pm z_{\alpha/2} \sqrt{\sum_i w_i^2(x) \hat{\sigma}^2(x)}$.

A better idea is to bootstrap the quantity

$$\frac{\sqrt{n} \sup_x |\hat{m}(x) - \mathbb{E}[\hat{m}(x)]|}{\hat{\sigma}(x)}$$

to get a bootstrap quantile t_n . Then

$$\left[\hat{m}(x) - \frac{t_n \hat{\sigma}(x)}{\sqrt{n}}, \hat{m}(x) + \frac{t_n \hat{\sigma}(x)}{\sqrt{n}} \right]$$

is a bootstrap variability band.

9 Wavelet smoothing

Not every nonparametric regression estimate needs to be a linear smoother (though this does seem to be very common), and *wavelet smoothing* is one of the leading nonlinear tools for nonparametric estimation. The theory of wavelets is elegant and we only give a brief introduction here; see [Mallat \(2008\)](#) for an excellent reference

You can think of wavelets as defining an orthonormal function basis, with the basis functions exhibiting a highly varied level of smoothness. Importantly, these basis functions also display spatially localized smoothness at different locations in the input domain. There are actually many different choices for wavelets bases (Haar wavelets, symmlets, etc.), but these are details that we will not go into

We assume $d = 1$. Local adaptivity in higher dimensions is not nearly as settled as it is with smoothing splines or (especially) kernels (multivariate extensions of wavelets are possible, i.e., *ridgelets* and *curvelets*, but are complex)

Consider basis functions, ϕ_1, \dots, ϕ_n , evaluated over n equally spaced inputs over $[0, 1]$:

$$X_i = i/n, \quad i = 1, \dots, n.$$

The assumption of evenly spaced inputs is crucial for fast computations; we also typically assume with wavelets that n is a power of 2. We now form a wavelet basis matrix $W \in \mathbb{R}^{n \times n}$, defined by

$$W_{ij} = \phi_j(X_i), \quad i, j = 1, \dots, n$$

The goal, given outputs $y = (y_1, \dots, y_n)$ over the evenly spaced input points, is to represent y as a sparse combination of the wavelet basis functions. To do so, we first perform a wavelet transform (multiply by W^T):

$$\tilde{\theta} = W^T y,$$

we threshold the coefficients θ (the threshold function T_λ to be defined shortly):

$$\hat{\theta} = T_\lambda(\tilde{\theta}),$$

and then perform an inverse wavelet transform (multiply by W):

$$\hat{\mu} = W\hat{\theta}$$

The wavelet and inverse wavelet transforms (multiplication by W^T and W) each require $O(n)$ operations, and are practically extremely fast due to clever pyramidal multiplication schemes that exploit the special structure of wavelets

The threshold function T_λ is usually taken to be hard-thresholding, i.e.,

$$[T_\lambda^{\text{hard}}(z)]_i = z_i \cdot \mathbf{1}\{|z_i| \geq \lambda\}, \quad i = 1, \dots, n,$$

or soft-thresholding, i.e.,

$$[T_\lambda^{\text{soft}}(z)]_i = (z_i - \text{sign}(z_i)\lambda) \cdot \mathbf{1}\{|z_i| \geq \lambda\}, \quad i = 1, \dots, n.$$

These thresholding functions are both also $O(n)$, and computationally trivial, making wavelet smoothing very fast overall

We should emphasize that wavelet smoothing is not a linear smoother, i.e., there is no single matrix S such that $\hat{\mu} = Sy$ for all y

We can write the wavelet smoothing estimate in a more familiar form, following our previous discussions on basis functions and regularization. For hard-thresholding, we solve

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \|y - W\theta\|_2^2 + \lambda^2 \|\theta\|_0,$$

and then the wavelet smoothing fitted values are $\hat{\mu} = W\hat{\theta}$. Here $\|\theta\|_0 = \sum_{i=1}^n \mathbf{1}\{\theta_i \neq 0\}$, the number of nonzero components of θ , called the “ ℓ_0 norm”. For soft-thresholding, we solve

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \|y - W\theta\|_2^2 + 2\lambda \|\theta\|_1,$$

and then the wavelet smoothing fitted values are $\hat{\mu} = W\hat{\theta}$. Here $\|\theta\|_1 = \sum_{i=1}^n |\theta_i|$, the ℓ_1 norm

9.1 The strengths of wavelets, the limitations of linear smoothers

Apart from its computational efficiency, an important strength of wavelet smoothing is that it can represent a signal that has a *spatially heterogeneous* degree of smoothness, i.e., it can be both smooth and wiggly at different regions of the input domain. The reason that wavelet smoothing can achieve such local adaptivity is because it selects a sparse number of wavelet basis functions, by thresholding the coefficients from a basis regression

We can make this more precise by considering convergence rates over an appropriate function class. In particular, we define the *total variation class* $M(k, C)$, for an integer $k \geq 0$ and $C > 0$, to contain all k times (weakly) differentiable functions whose k th derivative satisfies

$$\text{TV}(f^{(k)}) = \sup_{0=z_1 < z_2 < \dots < z_N < z_{N+1}=1} \sum_{j=1}^N |f^{(k)}(z_{i+1}) - f^{(k)}(z_i)| \leq C.$$

(Note that if f has $k+1$ continuous derivatives, then $\text{TV}(f^{(k)}) = \int_0^1 |f^{(k+1)}(x)| dx$.)

For the wavelet smoothing estimator, denoted by \hat{m}^{wav} , [Donoho & Johnstone \(1998\)](#) provide a seminal analysis. Assuming that $m_0 \in M(k, C)$ for a constant $C > 0$ (and further conditions on the setup), they show that (for an appropriate scaling of the smoothing parameter λ),

$$\mathbb{E}\|\hat{m}^{\text{wav}} - m_0\|_2^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{and} \quad \inf_{\hat{m}} \sup_{m_0 \in M(k, C)} \mathbb{E}\|\hat{m} - m_0\|_2^2 \gtrsim n^{-(2k+2)/(2k+3)}. \quad (21)$$

Thus wavelet smoothing attains the minimax optimal rate over the function class $M(k, C)$. (For a translation of this result to the notation of the current setting, see [Tibshirani \(2014\)](#).)

Some important questions: (i) just how big is the function class $M(k, C)$? And (ii) can a linear smoother also be minimax optimal over $M(k, C)$?

It is not hard to check $M(k, C) \supseteq S_1(k + 1, C')$, the (univariate) Sobolev space of order $k + 1$, for some other constant $C' > 0$. We know from the previously mentioned theory on Sobolev spaces that the minimax rate over $S_1(k + 1, C')$ is again $n^{-(2k+2)/(2k+3)}$. This suggests that these two function spaces might actually be somewhat close in size

But in fact, the overall minimax rates here are sort of misleading, and we will see from the behavior of linear smoothers that the function classes are actually quite different. [Donoho & Johnstone \(1998\)](#) showed that the minimax error over $M(k, C)$, *restricted to linear smoothers*, satisfies

$$\inf_{\hat{m} \text{ linear}} \sup_{m_0 \in M(k, C)} \mathbb{E}\|\hat{m} - m_0\|_2^2 \gtrsim n^{-(2k+1)/(2k+2)}. \quad (22)$$

(See again [Tibshirani \(2014\)](#) for a translation to the notation of the current setting.) Hence the answers to our questions are: (ii) linear smoothers cannot cope with the heterogeneity of functions in $M(k, C)$, and are bounded away from optimality, which means (i) we can interpret $M(k, C)$ as being much larger than $S_1(k + 1, C')$, because linear smoothers can be optimal over the latter class but not over the former. See Figure 5 for a diagram

Let's back up to emphasize just how remarkable the results (21), (22) really are. Though it may seem like a subtle difference in exponents, there is actually a significant difference in the minimax rate and minimax linear rate: e.g., when $k = 0$, this is a difference of $n^{-1/2}$ (optimal) and $n^{-1/2}$ (optimal among linear smoothers) for estimating a function of bounded variation. Recall also just how broad the linear smoother class is: kernel smoothing, regression splines, smoothing splines, RKHS estimators ... none of these methods can achieve a better rate than $n^{-1/2}$ over functions of bounded variation

Practically, the differences between wavelets and linear smoothers in problems with spatially heterogeneous smoothness can be striking as well. However, you should keep in mind that wavelets are not perfect: a shortcoming is that they require a highly restrictive setup: recall that they require evenly spaced inputs, and n to be power of 2, and there are often further assumptions made about the behavior of the fitted function at the boundaries of the input domain

Also, though you might say they marked the beginning of the story, wavelets are not the end of the story when it comes to local adaptivity. The natural thing to do, it might seem, is to make (say) kernel smoothing or smoothing splines more locally adaptive by allowing for a local bandwidth parameter or a local penalty parameter. People have tried this, but it

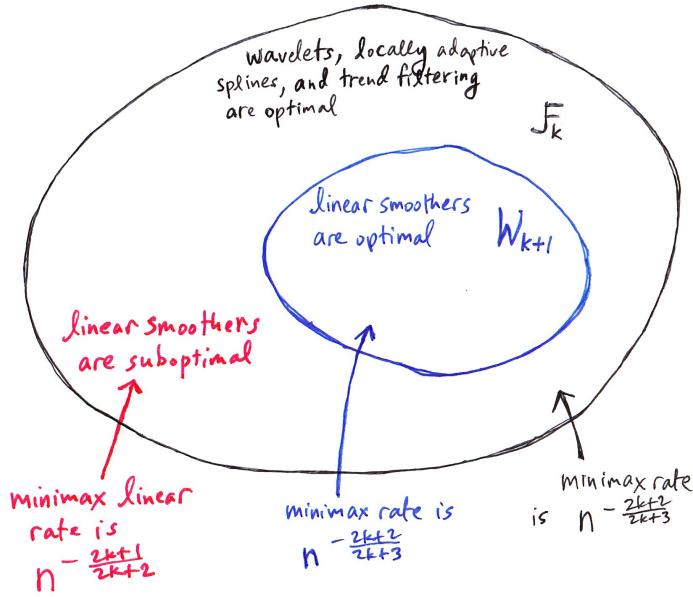


Figure 5: A diagram of the minimax rates over $M(k, C)$ (denoted \mathcal{F}_k in the picture) and $S_1(k + 1, C)$ (denoted \mathcal{W}_{k+1} in the picture)

is both difficult theoretically and practically to get right. A cleaner approach is to redesign the kind of penalization used in constructing smoothing splines directly.

10 More on Splines: Regression and Smoothing Splines

10.1 Splines

- Regression splines and smoothing splines are motivated from a different perspective than kernels and local polynomials; in the latter case, we started off with a special kind of local averaging, and moved our way up to a higher-order local models. With regression splines and smoothing splines, we build up our estimate globally, from a set of select basis functions
- These basis functions, as you might guess, are *splines*. Let's assume that $d = 1$ for simplicity. (We'll stay in the univariate case, for the most part, in this section.) A k th-order spline f is a piecewise polynomial function of degree k that is continuous and has continuous derivatives of orders $1, \dots, k - 1$, at its knot points. Specifically, there are $t_1 < \dots < t_p$ such that f is a polynomial of degree k on each of the intervals

$$(-\infty, t_1], [t_1, t_2], \dots, [t_p, \infty)$$

and $f^{(j)}$ is continuous at t_1, \dots, t_p , for each $j = 0, 1, \dots, k - 1$

- Splines have some special (some might say: amazing!) properties, and they have been a topic of interest among statisticians and mathematicians for a very long time. See

[de Boor \(1978\)](#) for an in-depth coverage. Informally, a spline is a lot smoother than a piecewise polynomial, and so modeling with splines can serve as a way of reducing the variance of fitted estimators. See Figure 6

- A bit of statistical folklore: it is said that a cubic spline is so smooth, that one cannot detect the locations of its knots by eye!
- How can we parametrize the set of a splines with knots at t_1, \dots, t_p ? The most natural way is to use the *truncated power basis*, g_1, \dots, g_{p+k+1} , defined as

$$\begin{aligned} g_1(x) &= 1, \quad g_2(x) = x, \quad \dots \quad g_{k+1}(x) = x^k, \\ g_{k+1+j}(x) &= (x - t_j)_+^k, \quad j = 1, \dots, p. \end{aligned} \tag{23}$$

(Here x_+ denotes the positive part of x , i.e., $x_+ = \max\{x, 0\}$.) From this we can see that the space of k th-order splines with knots at t_1, \dots, t_p has dimension $p + k + 1$

- While these basis functions are natural, a much better computational choice, both for speed and numerical accuracy, is the *B-spline* basis. This was a major development in spline theory and is now pretty much the standard in software. The key idea: B-splines have local support, so a basis matrix that we form with them (to be defined below) is banded. See [de Boor \(1978\)](#) or the Appendix of Chapter 5 in [Hastie et al. \(2009\)](#) for details

10.2 Regression splines

- A first idea: let's perform regression on a spline basis. In other words, given inputs x_1, \dots, x_n and responses y_1, \dots, y_n , we consider fitting functions f that are k th-order splines with knots at some chosen locations t_1, \dots, t_p . This means expressing f as

$$f(x) = \sum_{j=1}^{p+k+1} \beta_j g_j(x),$$

where $\beta_1, \dots, \beta_{p+k+1}$ are coefficients and g_1, \dots, g_{p+k+1} , are basis functions for order k splines over the knots t_1, \dots, t_p (e.g., the truncated power basis or B-spline basis)

- Letting $y = (y_1, \dots, y_n) \in \mathbb{R}^n$, and defining the basis matrix $G \in \mathbb{R}^{n \times (p+k+1)}$ by

$$G_{ij} = g_j(x_i), \quad i = 1, \dots, n, \quad j = 1, \dots, p + k + 1,$$

we can just use least squares to determine the optimal coefficients $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_{p+k+1})$,

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^{p+k+1}}{\operatorname{argmin}} \|y - G\beta\|_2^2,$$

which then leaves us with the fitted *regression spline* $\hat{f}(x) = \sum_{j=1}^{p+k+1} \hat{\beta}_j g_j(x)$

- Of course we know that $\hat{\beta} = (G^T G)^{-1} G^T y$, so the fitted values $\hat{\mu} = (\hat{f}(x_1), \dots, \hat{f}(x_n))$ are

$$\hat{\mu} = G(G^T G)^{-1} G^T y,$$

and regression splines are linear smoothers

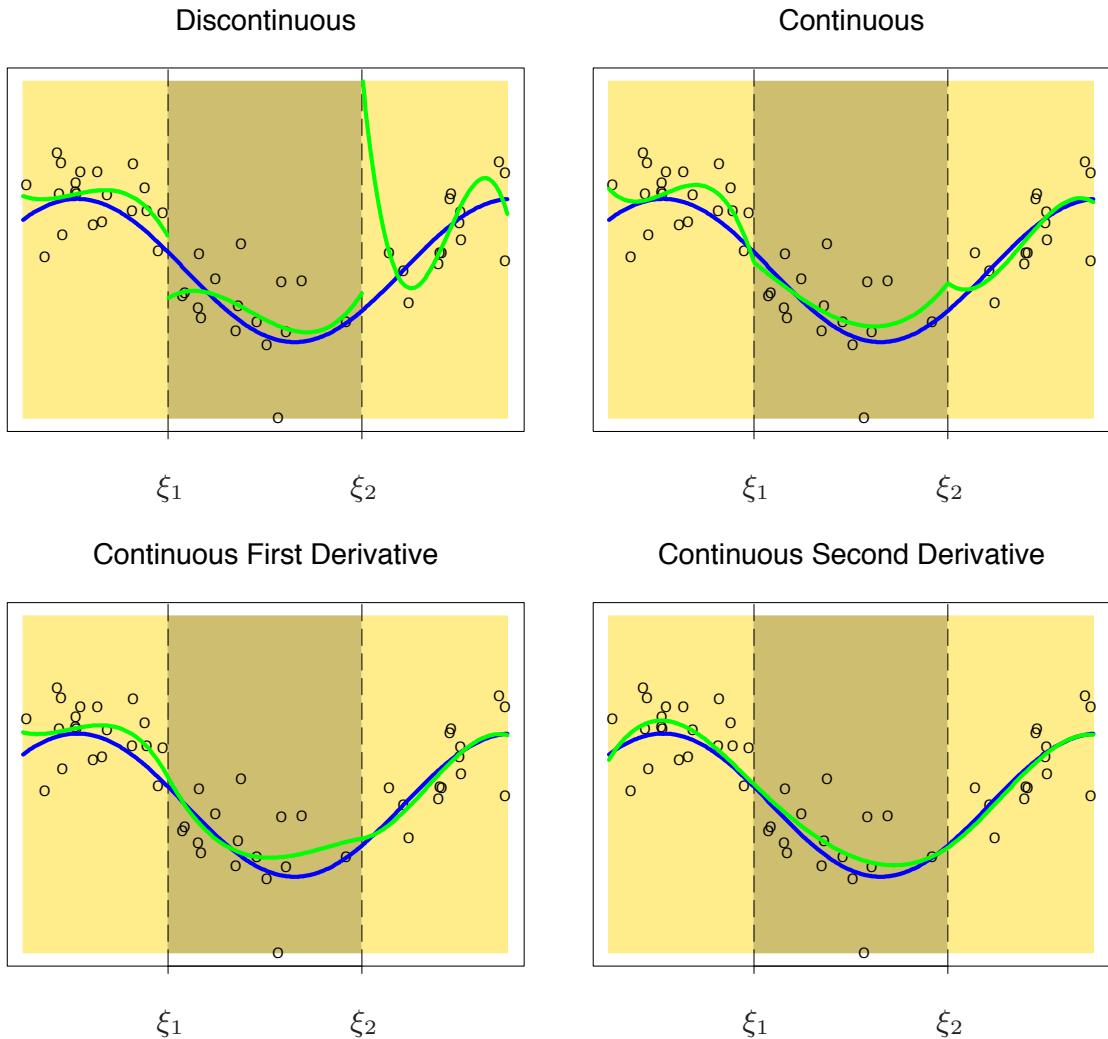


Figure 6: Illustration of the effects of enforcing continuity at the knots, across various orders of the derivative, for a cubic piecewise polynomial. From Chapter 5 of [Hastie et al. \(2009\)](#)

- This is a classic method, and can work well provided we choose good knots t_1, \dots, t_p ; but in general choosing knots is a tricky business. There is a large literature on knot selection for regression splines via greedy methods like recursive partitioning

10.3 Natural splines

- A problem with regression splines is that the estimates tend to display erratic behavior, i.e., they have high variance, at the boundaries of the input domain. (This is the opposite problem to that with kernel smoothing, which had poor bias at the boundaries.) This only gets worse as the polynomial order k gets larger
- A way to remedy this problem is to force the piecewise polynomial function to have a lower degree to the left of the leftmost knot, and to the right of the rightmost knot—this is exactly what *natural splines* do. A natural spline of order k , with knots at $t_1 < \dots < t_p$, is a piecewise polynomial function f such that
 - f is a polynomial of degree k on each of $[t_1, t_2], \dots, [t_{p-1}, t_p]$,
 - f is a polynomial of degree $(k-1)/2$ on $(-\infty, t_1]$ and $[t_p, \infty)$,
 - f is continuous and has continuous derivatives of orders $1, \dots, k-1$ at t_1, \dots, t_p .

It is implicit here that natural splines are only defined for odd orders k

- What is the dimension of the span of k th order natural splines with knots at t_1, \dots, t_p ? Recall for splines, this was $p+k+1$ (the number of truncated power basis functions). For natural splines, we can compute this dimension by counting:

$$\underbrace{(k+1) \cdot (p-1)}_a + \underbrace{\left(\frac{k-1}{2} + 1\right) \cdot 2}_b - \underbrace{k \cdot p}_c = p.$$

Above, a is the number of free parameters in the interior intervals $[t_1, t_2], \dots, [t_{p-1}, t_p]$, b is the number of free parameters in the exterior intervals $(-\infty, t_1], [t_p, \infty)$, and c is the number of constraints at the knots t_1, \dots, t_p . The fact that the total dimension is p is amazing; this is independent of k !

- Note that there is a variant of the truncated power basis for natural splines, and a variant of the B-spline basis for natural splines. Again, B-splines are the preferred parametrization for computational speed and stability
- Natural splines of cubic order is the most common special case: these are smooth piecewise cubic functions, that are simply linear beyond the leftmost and rightmost knots

10.4 Smoothing splines

- Smoothing splines, at the end of the day, are given by a regularized regression over the natural spline basis, placing knots at all inputs x_1, \dots, x_n . They circumvent the problem of knot selection (as they just use the inputs as knots), and they control

for overfitting by shrinking the coefficients of the estimated function (in its basis expansion)

- Interestingly, we can motivate and define a smoothing spline directly from a functional minimization perspective. With inputs x_1, \dots, x_n lying in an interval $[0, 1]$, the *smoothing spline* estimate \hat{f} , of a given odd integer order $k \geq 0$, is defined as

$$\hat{f} = \operatorname{argmin}_f \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_0^1 (f^{(m)}(x))^2 dx, \quad \text{where } m = (k+1)/2. \quad (24)$$

This is an infinite-dimensional optimization problem over all functions f for which the criterion is finite. This criterion trades off the least squares error of f over the observed pairs (x_i, y_i) , $i = 1, \dots, n$, with a penalty term that is large when the m th derivative of f is wiggly. The tuning parameter $\lambda \geq 0$ governs the strength of each term in the minimization

- By far the most commonly considered case is $k = 3$, i.e., cubic smoothing splines, which are defined as

$$\hat{f} = \operatorname{argmin}_f \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_0^1 f''(x)^2 dx \quad (25)$$

- Remarkably, it so happens that the minimizer in the general smoothing spline problem (38) is unique, and is a natural k th-order spline with knots at the input points x_1, \dots, x_n ! Here we give a proof for the cubic case, $k = 3$, from [Green & Silverman \(1994\)](#) (see also Exercise 5.7 in [Hastie et al. \(2009\)](#))

The key result can be stated as follows: if \tilde{f} is any twice differentiable function on $[0, 1]$, and $x_1, \dots, x_n \in [0, 1]$, then there exists a natural cubic spline f with knots at x_1, \dots, x_n such that $f(x_i) = \tilde{f}(x_i)$, $i = 1, \dots, n$ and

$$\int_0^1 f''(x)^2 dx \leq \int_0^1 \tilde{f}''(x)^2 dx.$$

Note that this would in fact prove that we can restrict our attention in (25) to natural splines with knots at x_1, \dots, x_n

Proof: the natural spline basis with knots at x_1, \dots, x_n is n -dimensional, so given any n points $z_i = \tilde{f}(x_i)$, $i = 1, \dots, n$, we can always find a natural spline f with knots at x_1, \dots, x_n that satisfies $f(x_i) = z_i$, $i = 1, \dots, n$. Now define

$$h(x) = \tilde{f}(x) - f(x).$$

Consider

$$\begin{aligned}
\int_0^1 f''(x)h''(x) dx &= f''(x)h'(x)\Big|_0^1 - \int_0^1 f'''(x)h'(x) dx \\
&= - \int_{x_1}^{x_n} f'''(x)h'(x) dx \\
&= - \sum_{j=1}^{n-1} f'''(x)h(x)\Big|_{x_j}^{x_{j+1}} + \int_{x_1}^{x_n} f^{(4)}(x)h'(x) dx \\
&= - \sum_{j=1}^{n-1} f'''(x_j^+)(h(x_{j+1}) - h(x_j)),
\end{aligned}$$

where in the first line we used integration by parts; in the second we used the fact that $f''(a) = f''(b) = 0$, and $f'''(x) = 0$ for $x \leq x_1$ and $x \geq x_n$, as f is a natural spline; in the third we used integration by parts again; in the fourth line we used the fact that f''' is constant on any open interval (x_j, x_{j+1}) , $j = 1, \dots, n-1$, and that $f^{(4)} = 0$, again because f is a natural spline. (In the above, we use $f'''(u^+)$ to denote $\lim_{x \downarrow u} f'''(x)$.) Finally, since $h(x_j) = 0$ for all $j = 1, \dots, n$, we have

$$\int_0^1 f''(x)h''(x) dx = 0.$$

From this, it follows that

$$\begin{aligned}
\int_0^1 \tilde{f}''(x)^2 dx &= \int_0^1 (f''(x) + h''(x))^2 dx \\
&= \int_0^1 f''(x)^2 dx + \int_0^1 h''(x)^2 dx + 2 \int_0^1 f''(x)h''(x) dx \\
&= \int_0^1 f''(x)^2 dx + \int_0^1 h''(x)^2 dx,
\end{aligned}$$

and therefore

$$\int_0^1 f''(x)^2 dx \leq \int_0^1 \tilde{f}''(x)^2 dx, \quad (26)$$

with equality if and only if $h''(x) = 0$ for all $x \in [0, 1]$. Note that $h'' = 0$ implies that h must be linear, and since we already know that $h(x_j) = 0$ for all $j = 1, \dots, n$, this is equivalent to $h = 0$. In other words, the inequality (45) holds strictly except when $\tilde{f} = f$, so the solution in (25) is uniquely a natural spline with knots at the inputs

10.5 Finite-dimensional form

- The key result presented above tells us that we can choose a basis η_1, \dots, η_n for the set of k th-order natural splines with knots over x_1, \dots, x_n , and reparametrize the problem (38) as

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^n \left(y_i - \sum_{j=1}^n \beta_j \eta_j(x_i) \right)^2 + \lambda \int_0^1 \left(\sum_{j=1}^n \beta_j \eta_j^{(m)}(x) \right)^2 dx. \quad (27)$$

This is a finite-dimensional problem, and after we compute the coefficients $\hat{\beta} \in \mathbb{R}^n$, we know that the smoothing spline estimate is simply $\hat{f}(x) = \sum_{j=1}^n \hat{\beta}_j \eta_j(x)$

- Defining the basis matrix and penalty matrices $N, \Omega \in \mathbb{R}^{n \times n}$ by

$$N_{ij} = \eta_j(x_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 \eta_i^{(m)}(x) \eta_j^{(m)}(x) dx \quad \text{for } i, j = 1, \dots, n, \quad (28)$$

the problem in (27) can be written more succinctly as

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \|y - N\beta\|_2^2 + \lambda \beta \Omega \beta, \quad (29)$$

showing the smoothing spline problem to be a type of generalized ridge regression problem. In fact, the solution in (29) has the explicit form

$$\hat{\beta} = (N^T N + \lambda \Omega)^{-1} N^T y,$$

and therefore the fitted values $\hat{\mu} = (\hat{f}(x_1), \dots, \hat{f}(x_n))$ are

$$\hat{\mu} = N(N^T N + \lambda \Omega)^{-1} N^T y. \quad (30)$$

Therefore, once again, smoothing splines are a type of linear smoother

- A special property of smoothing splines: the fitted values in (30) can be computed in $O(n)$ operations. This is achieved by forming N from the B-spline basis (for natural splines), and in this case the matrix $N^T N + \Omega I$ ends up being banded (with a bandwidth that only depends on the polynomial order k). In practice, smoothing spline computations are extremely fast

10.6 Reinsch form

- It is informative to rewrite the fitted values in (30) in what is called Reinsch form,

$$\begin{aligned} \hat{\mu} &= N(N^T N + \lambda \Omega)^{-1} N^T y \\ &= N \left(N^T (I + \lambda(N^T)^{-1} \Omega N^{-1}) N \right)^{-1} N^T y \\ &= (I + \lambda Q)^{-1} y, \end{aligned} \quad (31)$$

where $Q = (N^T)^{-1} \Omega N^{-1}$

- Note that this matrix Q does not depend on λ . If we compute an eigendecomposition $Q = UDU^T$, then the eigendecomposition of $S = N(N^T N + \lambda \Omega)^{-1} = (I + \lambda Q)^{-1}$ is

$$S = \sum_{j=1}^n \frac{1}{1 + \lambda d_j} u_j u_j^T,$$

where $D = \operatorname{diag}(d_1, \dots, d_n)$

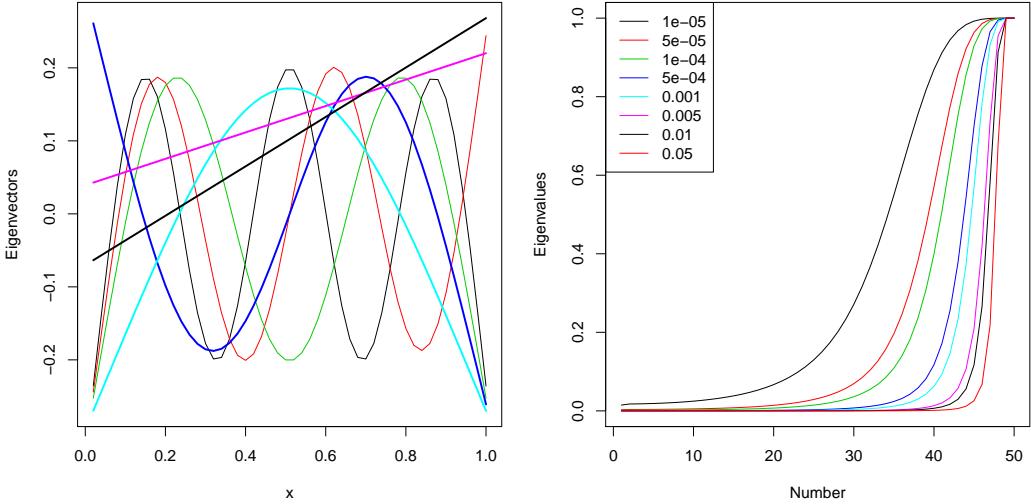


Figure 7: *Eigenvectors and eigenvalues for the Reinsch form of the cubic smoothing spline operator, defined over $n = 50$ evenly spaced inputs on $[0, 1]$. The left plot shows the bottom 7 eigenvectors of the Reinsch matrix Q . We can see that the smaller the eigenvalue, the “smoother” the eigenvector. The right plot shows the weights $w_j = 1/(1 + \lambda d_j)$, $j = 1, \dots, n$ implicitly used by the smoothing spline estimator (32), over 8 values of λ . We can see that when λ is larger, the weights decay faster, so the smoothing spline estimator places less weight on the “nonsmooth” eigenvectors*

- Therefore the smoothing spline fitted values are $\hat{\mu} = Sy$, i.e.,

$$\hat{\mu} = \sum_{j=1}^n \frac{u_j^T y}{1 + \lambda d_j} u_j. \quad (32)$$

Interpretation: smoothing splines perform a regression on the orthonormal basis $u_1, \dots, u_n \in \mathbb{R}^n$, yet they shrink the coefficients in this regression, with more shrinkage assigned to eigenvectors u_j that correspond to large eigenvalues d_j

- So what exactly are these basis vectors u_1, \dots, u_n ? These are known as the *Demmler-Reinsch basis*, and a lot of their properties can be worked out analytically (?). Basically: the eigenvectors u_j that correspond to smaller eigenvalues d_j are smoother, and so with smoothing splines, we shrink less in their direction. Said differently, by increasing λ in the smoothing spline estimator, we are tuning out the more wiggly components. See Figure 7

10.7 Kernel smoothing equivalence

- Something interesting happens when we plot the rows of the smoothing spline matrix S . For evenly spaced inputs, they look like the translations of a kernel! See Figure 8, left plot. For unevenly spaced inputs, the rows still have a kernel shape; now, the bandwidth appears to adapt to the density of the input points: lower density, larger bandwidth. See Figure 8, right plot

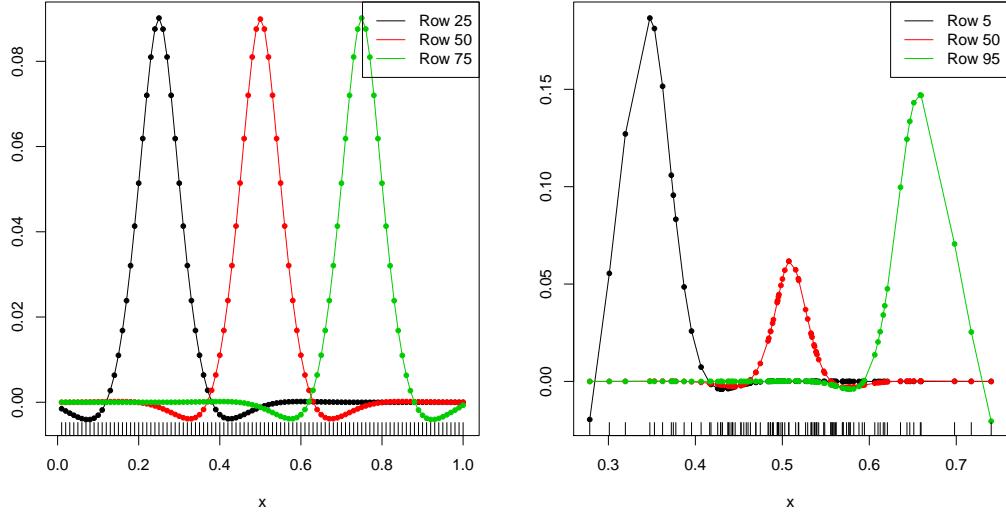


Figure 8: Rows of the cubic smoothing spline operator S defined over $n = 100$ evenly spaced input points on $[0, 1]$. The left plot shows 3 rows of S (in particular, rows 25, 50, and 75) for $\lambda = 0.0002$. These look precisely like translations of a kernel. The right plot considers a setup where the input points are concentrated around 0.5, and shows 3 rows of S (rows 5, 50, and 95) for the same value of λ . These still look like kernels, but the bandwidth is larger in low-density regions of the inputs

- What we are seeing is an empirical validation of a beautiful asymptotic result by ?. It turns out that the cubic smoothing spline estimator is asymptotically equivalent to a kernel regression estimator, with an unusual choice of kernel. Recall that both are linear smoothers; this equivalence is achieved by showing that under some conditions the smoothing spline weights converge to kernel weights, under the “Silverman kernel”:

$$K(x) = \frac{1}{2} \exp(-|x|/\sqrt{2}) \sin(|x|/\sqrt{2} + \pi/4), \quad (33)$$

and a local choice of bandwidth $h(x) = \lambda^{1/4}q(x)^{-1/4}$, where $q(x)$ is the density of the input points. That is, the bandwidth adapts to the local distribution of inputs. See Figure 9 for a plot of the Silverman kernel

- The Silverman kernel is “kind of” a higher-order kernel. It satisfies

$$\int K(x) dx = 1, \quad \int x^j K(x) dx = 0, \quad j = 1, \dots, 3, \quad \text{but} \quad \int x^4 K(x) dx = -24.$$

So it lies outside the scope of usual kernel analysis

- There is more recent work that connects smoothing splines of all orders to kernel smoothing. See, e.g., ??.

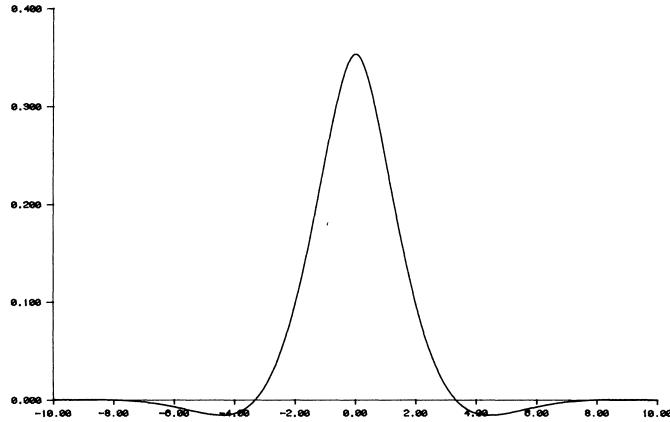


Figure 9: The Silverman kernel in (33), which is the (asymptotically) equivalent implicit kernel used by smoothing splines. Note that it can be negative. From ?

10.8 Error rates

- Define the Sobolev class of functions $W_1(m, C)$, for an integer $m \geq 0$ and $C > 0$, to contain all m times differentiable functions $f : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$\int (f^{(m)}(x))^2 dx \leq C^2.$$

(The Sobolev class $W_d(m, C)$ in d dimensions can be defined similarly, where we sum over all partial derivatives of order m .)

- Assuming $f_0 \in W_1(m, C)$ for the underlying regression function, where $C > 0$ is a constant, the smoothing spline estimator \hat{f} in (38) of polynomial order $k = 2m - 1$ with tuning parameter $\lambda \asymp n^{1/(2m+1)} \asymp n^{1/(k+2)}$ satisfies

$$\|\hat{f} - f_0\|_n^2 \lesssim n^{-2m/(2m+1)} \text{ in probability.}$$

The proof of this result uses much more fancy techniques from empirical process theory (entropy numbers) than the proofs for kernel smoothing. See Chapter 10.1 of [van de Geer \(2000\)](#)

- This rate is seen to be minimax optimal over $W_1(m, C)$ (e.g., [Nussbaum \(1985\)](#)). Also, it is worth noting that the Sobolev $W_1(m, C)$ and Holder $H_1(m, L)$ classes are equivalent in the following sense: given $W_1(m, C)$ for a constant $C > 0$, there are $L_0, L_1 > 0$ such that

$$H_1(m, L_0) \subseteq W_1(m, C) \subseteq H_1(m, L_1).$$

The first containment is easy to show; the second is far more subtle, and is a consequence of the Sobolev embedding theorem. (The same equivalences hold for the d -dimensional versions of the Sobolev and Holder spaces.)

10.9 Multivariate splines

- Splines can be extended to multiple dimensions, in two different ways: *thin-plate splines* and *tensor-product splines*. The former construction is more computationally efficient but more in some sense more limiting; the penalty for a thin-plate spline, of polynomial order $k = 2m - 1$, is

$$\sum_{\alpha_1+\dots+\alpha_d=m} \int \left| \frac{\partial^m f(x)}{\partial x_1^{\alpha_1} x_2^{\alpha_2} \dots \partial x_d^{\alpha_d}} \right|^2 dx,$$

which is rotationally invariant. Both of these concepts are discussed in Chapter 7 of [Green & Silverman \(1994\)](#) (see also Chapters 15 and 20.4 of [Gyorfi et al. \(2002\)](#))

- The multivariate extensions (thin-plate and tensor-product) of splines are highly non-trivial, especially when we compare them to the (conceptually) simple extension of kernel smoothing to higher dimensions. In multiple dimensions, if one wants to study penalized nonparametric estimation, it's (arguably) easier to study reproducing kernel Hilbert space estimators. We'll see, in fact, that this covers smoothing splines (and thin-plate splines) as a special case

References

- Bellman, R. (1962), *Adaptive Control Processes*, Princeton University Press.
- de Boor, C. (1978), *A Practical Guide to Splines*, Springer.
- Devroye, L., Gyorfi, L., & Lugosi, G. (1996), *A Probabilistic Theory of Pattern Recognition*, Springer.
- Donoho, D. L. & Johnstone, I. (1998), ‘Minimax estimation via wavelet shrinkage’, *Annals of Statistics* **26**(8), 879–921.
- Fan, J. (1993), ‘Local linear regression smoothers and their minimax efficiencies’, *The Annals of Statistics* pp. 196–216.
- Fan, J. & Gijbels, I. (1996), *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, Vol. 66, CRC Press.
- Green, P. & Silverman, B. (1994), *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman & Hall/CRC Press.
- Gyorfi, L., Kohler, M., Krzyzak, A. & Walk, H. (2002), *A Distribution-Free Theory of Nonparametric Regression*, Springer.
- Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall.
- Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer. Second edition.
- Johnstone, I. (2011), *Gaussian estimation: Sequence and wavelet models*, Under contract to Cambridge University Press. Online version at <http://www-stat.stanford.edu/~imj>.
- Kim, S.-J., Koh, K., Boyd, S. & Gorinevsky, D. (2009), ‘ ℓ_1 trend filtering’, *SIAM Review* **51**(2), 339–360.
- Lin, Y. & Zhang, H. H. (2006), ‘Component selection and smoothing in multivariate non-parametric regression’, *Annals of Statistics* **34**(5), 2272–2297.
- Mallat, S. (2008), *A wavelet tour of signal processing*, Academic Press. Third edition.
- Mammen, E. & van de Geer, S. (1997), ‘Locally adaptive regression splines’, *Annals of Statistics* **25**(1), 387–413.
- Nussbaum, M. (1985), ‘Spline smoothing in regression models and asymptotic efficiency in l_2 ’, *Annals of Statistics* **13**(3), 984–997.
- Raskutti, G., Wainwright, M. & Yu, B. (2012), ‘Minimax-optimal rates for sparse additive models over kernel classes via convex programming’, *Journal of Machine Learning Research* **13**, 389–427.
- Ravikumar, P., Liu, H., Lafferty, J. & Wasserman, L. (2009), ‘Sparse additive models’, *Journal of the Royal Statistical Society: Series B* **75**(1), 1009–1030.

- Scholkopf, B. & Smola, A. (2002), ‘Learning with kernels’.
- Simonoff, J. (1996), *Smoothing Methods in Statistics*, Springer.
- Steidl, G., Didas, S. & Neumann, J. (2006), ‘Splines in higher order TV regularization’, *International Journal of Computer Vision* **70**(3), 214–255.
- Stone, C. (1985), ‘Additive regression models and other nonparametric models’, *Annals of Statistics* **13**(2), 689–705.
- Tibshirani, R. J. (2014), ‘Adaptive piecewise polynomial estimation via trend filtering’, *Annals of Statistics* **42**(1), 285–323.
- Tsybakov, A. (2009), *Introduction to Nonparametric Estimation*, Springer.
- van de Geer, S. (2000), *Empirical Processes in M-Estimation*, Cambridge University Press.
- Wahba, G. (1990), *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics.
- Wang, Y., Smola, A. & Tibshirani, R. J. (2014), ‘The falling factorial basis and its statistical properties’, *International Conference on Machine Learning* **31**.
- Wasserman, L. (2006), *All of Nonparametric Statistics*, Springer.
- Yang, Y. (1999), ‘Nonparametric classification–Part I: Rates of convergence’, *IEEE Transactions on Information Theory* **45**(7), 2271–2284.

Appendix: Locally adaptive estimators

10.10 Locally adaptive regression splines

Locally adaptive regression splines (Mammen & van de Geer 1997), as their name suggests, can be viewed as variant of smoothing splines that exhibit better local adaptivity. For a given integer order $k \geq 0$, the estimate is defined as

$$\hat{m} = \operatorname{argmin}_f \sum_{i=1}^n (Y_i - m(X_i))^2 + \lambda \operatorname{TV}(f^{(k)}). \quad (34)$$

The minimization domain is infinite-dimensional, the space of all functions for which the criterion is finite

Another remarkable variational result, similar to that for smoothing splines, shows that (34) has a k th order spline as a solution (Mammen & van de Geer 1997). This *almost* turns the minimization into a finite-dimensional one, but there is one catch: the knots of this k th-order spline are generally not known, i.e., they need not coincide with the inputs x_1, \dots, x_n . (When $k = 0, 1$, they do, but in general, they do not)

To deal with this issue, we can redefine the locally adaptive regression spline estimator to be

$$\hat{m} = \operatorname{argmin}_{f \in \mathcal{G}_k} \sum_{i=1}^n (Y_i - m(X_i))^2 + \lambda \operatorname{TV}(f^{(k)}), \quad (35)$$

i.e., we restrict the domain of minimization to be \mathcal{G}_k , the space of k th-order spline functions with knots in T_k , where T_k is a subset of $\{x_1, \dots, x_n\}$ of size $n - k - 1$. The precise definition of T_k is not important; it is just given by trimming away $k + 1$ boundary points from the inputs

As we already know, the space \mathcal{G}_k of k th-order splines with knots in T_k has dimension $|T_k| + k + 1 = n$. Therefore we can choose a basis g_1, \dots, g_n for the functions in \mathcal{G}_k , and the problem in (35) becomes one of finding the coefficients in this basis expansion,

$$\hat{\beta} = \operatorname{argmin}_{f \in \mathcal{G}_k} \sum_{i=1}^n \left(Y_i - \sum_{j=1}^n \beta_j g_j(X_i) \right)^2 + \lambda \operatorname{TV} \left\{ \left(\sum_{j=1}^n \beta_j g_j(X_i) \right)^{(k)} \right\}, \quad (36)$$

and then we have $\hat{m}(x) = \sum_{j=1}^n \hat{\beta}_j g_j(x)$

Now define the basis matrix $G \in \mathbb{R}^{n \times n}$ by

$$G_{ij} = g_j(X_i), \quad i = 1, \dots, n.$$

Suppose we choose g_1, \dots, g_n to be the truncated power basis. Denoting $T_k = \{t_1, \dots, t_{n-k-1}\}$, we compute

$$\left(\sum_{j=1}^n \beta_j g_j(X_i) \right)^{(k)} = k. + k. \sum_{j=k+2}^n \beta_j 1\{x \geq t_{j-k-1}\},$$

and so

$$\operatorname{TV} \left\{ \left(\sum_{j=1}^n \beta_j g_j(X_i) \right)^{(k)} \right\} = k. \sum_{j=k+2}^n |\beta_j|.$$

Hence the locally adaptive regression spline problem (36) can be expressed as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \|y - G\beta\|_2^2 + \lambda k \sum_{i=k+2}^n |\beta_i|. \quad (37)$$

This is a lasso regression problem on the truncated power basis matrix G , with the first $k+1$ coefficients (those corresponding to the pure polynomial functions, in the basis expansion) left unpenalized

This reveals a key difference between the locally adaptive regression splines (37) (originally, problem (35)) and the smoothing splines (29) (originally, problem

$$\widehat{m} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^n (Y_i - m(X_i))^2 + \lambda \int_0^1 (f^{(m)}(x))^2 dx, \quad \text{where } m = (k+1)/2. \quad (38)$$

In the first problem, the total variation penalty is translated into an ℓ_1 penalty on the coefficients of the truncated power basis, and hence this acts a knot selector for the estimated function. That is, at the solution in (37), the estimated spline has knots at a subset of T_k (at a subset of the input points x_1, \dots, x_n), with fewer knots when λ is larger. In contrast, recall, at the smoothing spline solution in (29), the estimated function has knots at each of the inputs x_1, \dots, x_n . This is a major difference between the ℓ_1 and ℓ_2 penalties

From a computational perspective, the locally adaptive regression spline problem in (37) is actually a lot harder than the smoothing spline problem in (29). Recall that the latter reduces to solving a single banded linear system, which takes $O(n)$ operations. On the other hand, fitting locally adaptive regression splines in (37) requires solving a lasso problem with a dense $n \times n$ regression matrix G ; this takes something like $O(n^3)$ operations. So when $n = 10,000$, there is a big difference between the two.

There is a tradeoff here, as with extra computation comes much improved local adaptivity of the fits. See Figure 10 for an example. Theoretically, when $m_0 \in M(k, C)$ for a constant $C > 0$, Mammen & van de Geer (1997) show the locally adaptive regression spline estimator, denoted \widehat{m}^{trs} , with $\lambda \asymp n^{1/(2k+3)}$, satisfies

$$\|\widehat{m}^{\text{trs}} - m_0\|_n^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{in probability,}$$

so (like wavelets) it achieves the minimax optimal rate over $n^{-(2k+2)/(2k+3)}$. In this regard, as we discussed previously, they actually have a big advantage over any linear smoother (not just smoothing splines)

10.11 Trend filtering

At a high level, you can think of trend filtering as computationally efficient version of locally adaptive regression splines, though their original construction (Steidl et al. 2006, Kim et al. 2009) comes from a fairly different perspective. We will begin by describing their connection to locally adaptive regression splines, following Tibshirani (2014)

Revisit the formulation of locally adaptive regression splines in (35), where the minimization domain is $\mathcal{G}_k = \text{span}\{g_1, \dots, g_n\}$, and g_1, \dots, g_n are the k th-order truncated power basis

$$\begin{aligned} g_1(x) &= 1, \quad g_2(x) = x, \quad \dots \quad g_{k+1}(x) = x^k, \\ g_{k+1+j}(x) &= (x - t_j)_+^k, \quad j = 1, \dots, p. \end{aligned} \quad (39)$$

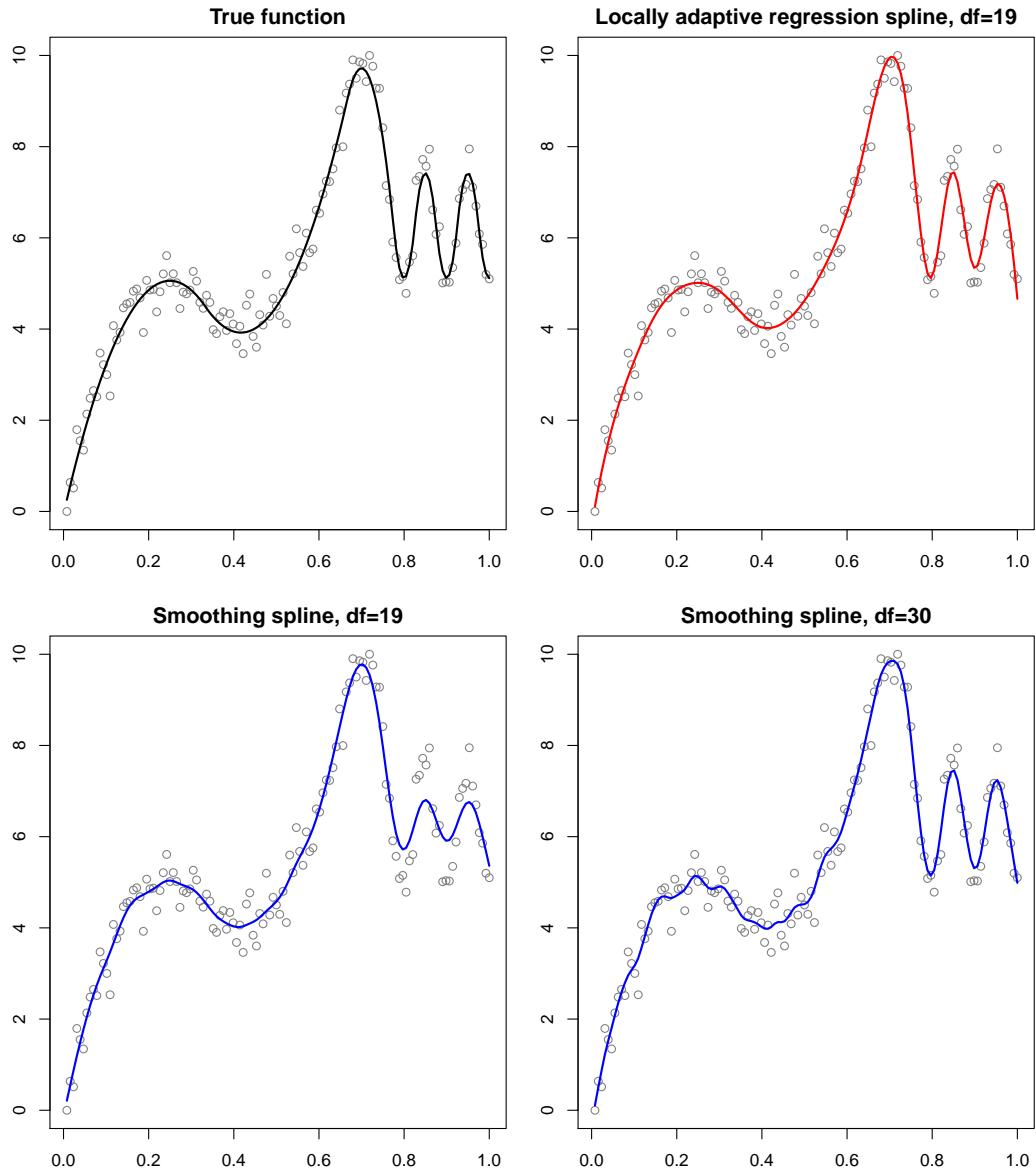


Figure 10: The top left plot shows a simulated true regression function, which has inhomogeneous smoothness: smoother towards the left part of the domain, wigglier towards the right. The top right plot shows the locally adaptive regression spline estimate with 19 degrees of freedom; notice that it picks up the right level of smoothness throughout. The bottom left panel shows the smoothing spline estimate with the same degrees of freedom; it picks up the right level of smoothness on the left, but is undersmoothed on the right. The bottom right panel shows the smoothing spline estimate with 33 degrees of freedom; now it is appropriately wiggly on the right, but oversmoothed on the left. Smoothing splines cannot simultaneously represent different levels of smoothness at different regions in the domain; the same is true of any linear smoother

having knots in a set $T_k \subseteq \{X_1, \dots, X_n\}$ with size $|T_k| = n - k - 1$. The *trend filtering* problem is given by replacing \mathcal{G}_k with a different function space,

$$\hat{m} = \operatorname*{argmin}_{f \in \mathcal{H}_k} \sum_{i=1}^n (Y_i - m(X_i))^2 + \lambda \operatorname{TV}(f^{(k)}), \quad (40)$$

where the new domain is $\mathcal{H}_k = \text{span}\{h_1, \dots, h_n\}$. Assuming that the input points are ordered, $x_1 < \dots < x_n$, the functions h_1, \dots, h_n are defined by

$$h_j(x) = \prod_{\ell=1}^{j-1} (x - x_\ell), \quad j = 1, \dots, k+1,$$

$$h_{k+1+j}(x) = \prod_{\ell=1}^k (x - x_{j+\ell}) \cdot 1\{x \geq x_{j+k}\}, \quad j = 1, \dots, n-k-1. \quad (41)$$

(Our convention is to take the empty product to be 1, so that $h_1(x) = 1$.) These are dubbed the *falling factorial basis*, and are piecewise polynomial functions, taking an analogous form to the truncated power basis functions in (10.11). Loosely speaking, they are given by replacing an r th-order power function in the truncated power basis with an appropriate r -term product, e.g., replacing x^2 with $(x - x_2)(x - x_1)$, and $(x - t_j)^k$ with $(x - x_{j+k})(x - x_{j+k-1}) \cdots, (x - x_{j+1})$

Defining the falling factorial basis matrix

$$H_{ij} = h_j(X_i), \quad i, j = 1, \dots, n,$$

it is now straightforward to check that the proposed problem of study, trend filtering in (40), is equivalent to

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \|y - H\beta\|_2^2 + \lambda k \cdot \sum_{i=k+2}^n |\beta_i|. \quad (42)$$

This is still a lasso problem, but now in the falling factorial basis matrix H . Compared to the locally adaptive regression spline problem (37), there may not seem to be much of a difference here—like G , the matrix H is dense, and solving (42) would be slow. So why did we go to all the trouble of defining trend filtering, i.e., introducing the somewhat odd basis h_1, \dots, h_n in (41)?

The usefulness of trend filtering (42) is seen after reparametrizing the problem, by inverting H . Let $\theta = H\beta$, and rewrite the trend filtering problem as

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \|y - \theta\|_2^2 + \lambda \|D\theta\|_1, \quad (43)$$

where $D \in \mathbb{R}^{(n-k-1) \times n}$ denotes the last $n - k - 1$ rows of $k \cdot H^{-1}$. Explicit calculation shows that D is a banded matrix (Tibshirani 2014, Wang et al. 2014). For simplicity of exposition, consider the case when $X_i = i$, $i = 1, \dots, n$. Then, e.g., the first 3 orders of difference operators are:

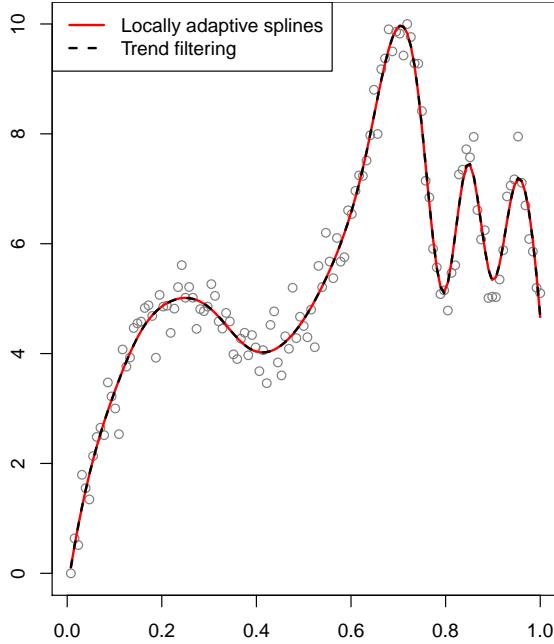


Figure 11: *Trend filtering and locally adaptive regression spline estimates, fit on the same data set as in Figure 10. The two are tuned at the same level, and the estimates are visually indistinguishable*

One can hence interpret D as a type of discrete derivative operator, of order $k + 1$. This also suggests an intuitive interpretation of trend filtering (43) as a discrete approximation to the original locally adaptive regression spline problem in (34)

The bandedness of D means that the trend filtering problem (43) can be solved efficiently, in close to linear time (complexity $O(n^{1.5})$ in the worst case). Thus trend filtering estimates are much easier to fit than locally adaptive regression splines

But what of their statistical relevancy? Did switching over to the falling factorial basis (41) wreck the local adaptivity properties that we cared about in the first place? Fortunately, the answer is no, and in fact, trend filtering and locally adaptive regression spline estimates are extremely hard to distinguish in practice. See Figure 11

Moreover, Tibshirani (2014), Wang et al. (2014) prove that the estimates from trend filtering and locally adaptive regression spline estimates, denoted \hat{m}^{tf} and \hat{m}^{trs} , respectively, when the tuning parameter λ for each scales as $n^{1/(2k+3)}$, satisfy

$$\|\hat{m}^{\text{tv}} - \hat{m}^{\text{trs}}\|_n^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{in probability.}$$

This coupling shows that trend filtering converges to the underlying function m_0 at the rate $n^{-(2k+2)/(2k+3)}$ whenever locally adaptive regression splines do, making them also minimax optimal over $M(k, C)$. In short, trend filtering offers provably significant improvements over linear smoothers, with a computational cost that is not too much steeper than a single banded linear system solve

10.12 Proof of (9)

Let

$$m_h(x) = \frac{\sum_{i=1}^n m(X_i)I(\|X_i - x\| \leq h)}{nP_n(B(x, h))}.$$

Let $A_n = \{P_n(B(x, h)) > 0\}$. When A_n is true,

$$\mathbb{E}\left((\hat{m}_h(x) - m_h(x))^2 \mid X_1, \dots, X_n\right) = \frac{\sum_{i=1}^n \text{Var}(Y_i | X_i)I(\|X_i - x\| \leq h)}{n^2 P_n^2(B(x, h))} \leq \frac{\sigma^2}{nP_n(B(x, h))}.$$

Since $m \in \mathcal{M}$, we have that $|m(X_i) - m(x)| \leq L\|X_i - x\| \leq Lh$ for $X_i \in B(x, h)$ and hence

$$|m_h(x) - m(x)|^2 \leq L^2 h^2 + m^2(x)I_{A_n(x)^c}.$$

Therefore,

$$\begin{aligned} \mathbb{E} \int (\hat{m}_h(x) - m(x))^2 dP(x) &= \mathbb{E} \int (\hat{m}_h(x) - m_h(x))^2 dP(x) + \mathbb{E} \int (m_h(x) - m(x))^2 dP(x) \\ &\leq \mathbb{E} \int \frac{\sigma^2}{nP_n(B(x, h))} I_{A_n(x)} dP(x) + L^2 h^2 + \int m^2(x) \mathbb{E}(I_{A_n(x)^c}) dP(x). \end{aligned} \quad (44)$$

To bound the first term, let $Y = nP_n(B(x, h))$. Note that $Y \sim \text{Binomial}(n, q)$ where $q = \mathbb{P}(X \in B(x, h))$. Now,

$$\begin{aligned} \mathbb{E} \left(\frac{I(Y > 0)}{Y} \right) &\leq \mathbb{E} \left(\frac{2}{1+Y} \right) = \sum_{k=0}^n \frac{2}{k+1} \binom{n}{k} q^k (1-q)^{n-k} \\ &= \frac{2}{(n+1)q} \sum_{k=0}^n \binom{n+1}{k+1} q^{k+1} (1-q)^{n-k} \\ &\leq \frac{2}{(n+1)q} \sum_{k=0}^{n+1} \binom{n+1}{k} q^k (1-q)^{n-k+1} \\ &= \frac{2}{(n+1)q} (q + (1-q))^{n+1} = \frac{2}{(n+1)q} \leq \frac{2}{nq}. \end{aligned}$$

Therefore,

$$\mathbb{E} \int \frac{\sigma^2 I_{A_n(x)}}{nP_n(B(x, h))} dP(x) \leq 2\sigma^2 \int \frac{dP(x)}{nP(B(x, h))}.$$

We may choose points z_1, \dots, z_M such that the support of P is covered by $\bigcup_{j=1}^M B(z_j, h/2)$ where $M \leq c_2/(nh^d)$. Thus,

$$\begin{aligned} \int \frac{dP(x)}{nP(B(x, h))} &\leq \sum_{j=1}^M \int \frac{I(z \in B(z_j, h/2))}{nP(B(x, h))} dP(x) \leq \sum_{j=1}^M \int \frac{I(z \in B(z_j, h/2))}{nP(B(z_j, h/2))} dP(x) \\ &\leq \frac{M}{n} \leq \frac{c_1}{nh^d}. \end{aligned}$$

The third term in (44) is bounded by

$$\begin{aligned}
\int m^2(x) \mathbb{E}(I_{A_n(x)^c}) dP(x) &\leq \sup_x m^2(x) \int (1 - P(B(x, h)))^n dP(x) \\
&\leq \sup_x m^2(x) \int e^{-nP(B(x, h))} dP(x) \\
&= \sup_x m^2(x) \int e^{-nP(B(x, h))} \frac{n P(B(x, h))}{nP(B(x, h))} dP(x) \\
&\leq \sup_x m^2(x) \sup_u (ue^{-u}) \int \frac{1}{nP(B(x, h))} dP(x) \\
&\leq \sup_x m^2(x) \sup_u (ue^{-u}) \frac{c_1}{nh^d} = \frac{c_2}{nh^d}.
\end{aligned}$$

10.13 Proof of the Spline Lemma

The key result can be stated as follows: if \tilde{f} is any twice differentiable function on $[0, 1]$, and $x_1, \dots, x_n \in [0, 1]$, then there exists a natural cubic spline f with knots at x_1, \dots, x_n such that $m(X_i) = \tilde{f}(X_i)$, $i = 1, \dots, n$ and

$$\int_0^1 f''(x)^2 dx \leq \int_0^1 \tilde{f}''(x)^2 dx.$$

Note that this would in fact prove that we can restrict our attention in (25) to natural splines with knots at x_1, \dots, x_n .

The natural spline basis with knots at x_1, \dots, x_n is n -dimensional, so given any n points $z_i = \tilde{f}(X_i)$, $i = 1, \dots, n$, we can always find a natural spline f with knots at x_1, \dots, x_n that satisfies $m(X_i) = z_i$, $i = 1, \dots, n$. Now define

$$h(x) = \tilde{f}(x) - m(x).$$

Consider

$$\begin{aligned}
\int_0^1 f''(x) h''(x) dx &= f''(x) h'(x) \Big|_0^1 - \int_0^1 f'''(x) h'(x) dx \\
&= - \int_{x_1}^{x_n} f'''(x) h'(x) dx \\
&= - \sum_{j=1}^{n-1} f'''(x) h(x) \Big|_{x_j}^{x_{j+1}} + \int_{x_1}^{x_n} f^{(4)}(x) h'(x) dx \\
&= - \sum_{j=1}^{n-1} f'''(x_j^+) (h(x_{j+1}) - h(x_j)),
\end{aligned}$$

where in the first line we used integration by parts; in the second we used the that $f''(a) = f''(b) = 0$, and $f'''(x) = 0$ for $x \leq x_1$ and $x \geq x_n$, as f is a natural spline; in the third we used integration by parts again; in the fourth line we used the fact that f''' is constant on any open interval (x_j, x_{j+1}) , $j = 1, \dots, n-1$, and that $f^{(4)} = 0$, again because f is a natural

spline. (In the above, we use $f'''(u^+)$ to denote $\lim_{x \downarrow u} f'''(x)$.) Finally, since $h(x_j) = 0$ for all $j = 1, \dots, n$, we have

$$\int_0^1 f''(x)h''(x) dx = 0.$$

From this, it follows that

$$\begin{aligned} \int_0^1 \tilde{f}''(x)^2 dx &= \int_0^1 (f''(x) + h''(x))^2 dx \\ &= \int_0^1 f''(x)^2 dx + \int_0^1 h''(x)^2 dx + 2 \int_0^1 f''(x)h''(x) dx \\ &= \int_0^1 f''(x)^2 dx + \int_0^1 h''(x)^2 dx, \end{aligned}$$

and therefore

$$\int_0^1 f''(x)^2 dx \leq \int_0^1 \tilde{f}''(x)^2 dx, \quad (45)$$

with equality if and only if $h''(x) = 0$ for all $x \in [0, 1]$. Note that $h'' = 0$ implies that h must be linear, and since we already know that $h(x_j) = 0$ for all $j = 1, \dots, n$, this is equivalent to $h = 0$. In other words, the inequality (45) holds strictly except when $\tilde{f} = f$, so the solution in (25) is uniquely a natural spline with knots at the inputs.

Linear Regression

We observe $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ where $X_i = (X_i(1), \dots, X_i(d)) \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}$. For notational simplicity, we will always assume that $X_i(1) = 1$.

Given a new pair (X, Y) we want to predict Y from X . The *conditional prediction risk* is

$$R(\hat{m}) = \mathbb{E}[(Y - \hat{m}(X))^2 | \mathcal{D}] = \int (y - \hat{m}(x))^2 dP(x, y)$$

and the *prediction risk* of \hat{m} is

$$r(\hat{m}) = \mathbb{E}(Y - \hat{m}(X))^2 = \mathbb{E}[r(\hat{m})]$$

where the expected value is over all random variables. The true regression function is

$$m(x) = \mathbb{E}[Y | X = x].$$

We have the following bias-variance decomposition:

$$r(\hat{m}) = \sigma^2 + \int b_n^2(x) dP(x) + \int v_n(x) dP(x)$$

where

$$\sigma^2 = \mathbb{E}[Y - m(X)]^2, \quad b_n(x) = \mathbb{E}[\hat{m}(x)] - m(x), \quad v_n(x) = \text{Var}(\hat{m}(x)).$$

Let $\epsilon = Y - m(X)$. Note that

$$\mathbb{E}[\epsilon] = \mathbb{E}[Y - m(X)] = \mathbb{E}[\mathbb{E}[Y - m(X) | X]] = 0.$$

A *linear predictor* has the form $g(x) = \beta^T x$. The *best linear predictor* minimizes $\mathbb{E}(Y - \beta^T X)^2$. (We do not assume that $m(x)$ is linear.) The minimizer, assuming that Σ is non-singular, is

$$\beta_* = \Sigma^{-1} \alpha$$

where $\Sigma = \mathbb{E}[XX^T]$ and $\alpha = \mathbb{E}(YX)$. **We will use linear predictors; but we should never assume that $m(x)$ is linear.** The *excess risk* is of the linear predictor $\beta^T x$ is

$$r(\beta) - r(\beta_*) = (\beta - \beta_*)^T \Sigma (\beta - \beta_*). \tag{1}$$

The *training error* is

$$\hat{r}_n(\beta) = \frac{1}{n} \sum_i (Y_i - X_i^T \beta)^2$$

1 Low Dimensional Linear Regression

Recall that $\Sigma = \mathbb{E}[XX^T]$. The *least squares estimator* $\widehat{\beta}$ minimizes the training error $\widehat{r}_n(\beta)$. We then have that

$$\widehat{\beta} = \widehat{\Sigma}^{-1}\widehat{\alpha}$$

where

$$\widehat{\Sigma} = \frac{1}{n} \sum_i X_i X_i^T, \quad \widehat{\alpha} = \frac{1}{n} \sum_i Y_i X_i.$$

We want to show that $r(\widehat{\beta})$ is close to $r(\beta_*)$. For simplicity, we will assume that the distribution P of (Y_i, X_i) supported on a compact set. Also, for simplicity, we assume that $\widehat{\beta}$ is truncated by some large constant L .

Theorem 1 *Let \mathcal{P} be the set of all distributions for $Z = (X, Y)$ supported on a compact set K . There exists constants c_1, c_2 such that the following is true. For any $\epsilon > 0$,*

$$\sup_{P \in \mathcal{P}} P^n \left(r(\widehat{\beta}_n) > r(\beta_*(P)) + 2\epsilon \right) \leq c_1 e^{-nc_2\epsilon^2}. \quad (2)$$

Hence,

$$r(\widehat{\beta}_n) - r(\beta_*) = O_P \left(\sqrt{\frac{1}{n}} \right).$$

Proof. Given any β , define $\widetilde{\beta} = (-1, \beta)$ and $\Lambda = \mathbb{E}[ZZ^T]$ where $Z = (Y, X)$. Note that

$$r(\beta) = \mathbb{E}(Y - \beta^T X)^2 = \mathbb{E}[(Z^T \widetilde{\beta})^2] = \widetilde{\beta}^T \Lambda \widetilde{\beta}.$$

Similarly,

$$\widehat{r}_n(\beta) = \widetilde{\beta}^T \widehat{\Lambda}_n \widetilde{\beta}$$

where

$$\widehat{\Lambda}_n = \frac{1}{n} \sum_i Z_i Z_i^T.$$

So

$$|\widehat{r}_n(\beta) - r(\beta)| = |\widetilde{\beta}^T (\widehat{\Lambda}_n - \Lambda) \widetilde{\beta}| \leq \|\widetilde{\beta}\|_1^2 \Delta_n$$

where

$$\Delta_n = \max_{j,k} |\widehat{\Lambda}_n(j, k) - \Lambda(j, k)|.$$

By Hoeffding's inequality and the union bound,

$$P \left(\sup_{\beta \in B} |\widehat{r}_n(\beta) - r(\beta)| > \epsilon \right) \leq c_1 e^{-nc_2\epsilon^2}.$$

On the event $\sup_{\beta \in B} |\widehat{r}_n(\beta) - r(\beta)| < \epsilon$, we have

$$r(\beta_*) \leq r(\widehat{\beta}_n) \leq \widehat{r}_n(\widehat{\beta}_n) + \epsilon \leq \widehat{r}_n(\beta_*) + \epsilon \leq r(\beta_*) + 2\epsilon.$$

□

The above result is not tight. Here is a more refined bound.

Theorem 2 (Theorem 11.3 of Gyorfi, Kohler, Krzyzak and Walk, 2002) *Let $\sigma^2 = \sup_x \text{Var}(Y|X = x) < \infty$. Assume that all the random variables are bounded by $L < \infty$. Then*

$$\mathbb{E} \int |\widehat{\beta}^T x - m(x)|^2 dP(x) \leq 8 \inf_{\beta} \int |\beta^T x - m(x)|^2 dP(x) + \frac{Cd(\log(n) + 1)}{n}.$$

The proof is straightforward but is very long. The strategy is to first bound $n^{-1} \sum_i (\widehat{\beta}^T X_i - m(X_i))^2$ using the properties of least squares. Then, using concentration of measure one can relate $n^{-1} \sum_i f^2(X_i)$ to $\int f^2(x) dP(x)$.

We have the following central limit theorem for $\widehat{\beta}$.

Theorem 3 *We have*

$$\sqrt{n}(\widehat{\beta} - \beta) \rightsquigarrow N(0, \Gamma)$$

where

$$\Gamma = \Sigma^{-1} \mathbb{E} \left[(Y - X^T \beta)^2 X X^T \right] \Sigma^{-1}$$

The covariance matrix Γ can be consistently estimated by

$$\widehat{\Gamma} = \widehat{\Sigma}^{-1} \widehat{M} \widehat{\Sigma}^{-1}$$

where

$$\widehat{M}(j, k) = \frac{1}{n} \sum_{i=1}^n X_i(j) X_i(k) \widehat{\epsilon}_i^2$$

and $\widehat{\epsilon}_i = Y_i - \widehat{\beta}^T X_i$.

The matrix $\widehat{\Gamma}$ is called the *sandwich estimator*. The Normal approximation can be used to construct confidence intervals for β . For example, $\widehat{\beta}(j) \pm z_\alpha \sqrt{\widehat{\Gamma}(j, j)/n}$ is an asymptotic $1 - \alpha$ confidence interval for $\beta(j)$. We can also get confidence intervals by using the bootstrap. Do **not** use the textbook formulas for the standard errors of $\widehat{\beta}$. These assume that the regression function itself is linear. See Buja et al (2015) for details.

2 High Dimensional Linear Regression

Now suppose that $d > n$. We can no longer use least squares. There are many approaches.

The simplest is to preprocess the data to reduce the dimension. For example, we can perform PCA on the X 's and use the first k principal components where $k < n$. Alternatively, we can cluster the covariates based on their correlations. We can then use one feature from each cluster or take the average of the covariates within each cluster. Another approach is to screen the variables by choosing the k features with the largest correlation with Y . After dimension reduction, we can then use least squares. These preprocessing methods can be very effective.

A different approach is to use all the covariates but, instead of least squares, we shrink the coefficients towards 0. This is called *ridge regression* and is discussed in the next section.

Yet another approach is model selection where we try to find a good subset of the covariates. Let S be a subset of $\{1, \dots, d\}$ and let $X_S = (X(j) : j \in S)$. If the size of S is not too large, we can regress Y on X_S instead of S .

In particular, fix $k < n$ and let \mathcal{S}_k denote all subsets of size k . For a given $S \in \mathcal{S}_k$, let β_S be the best linear predictor $\beta_S = \Sigma_S^{-1} \alpha_S$ for the subset S . We would like to choose $S \in \mathcal{S}_k$ to minimize

$$\mathbb{E}(Y - \beta_S^T X_S)^2.$$

This is equivalent to:

$$\text{minimize } \mathbb{E}(Y - \beta^T X)^2 \quad \text{subject to } \|\beta\|_0 \leq k$$

where $\|\beta\|_0$ is the number of non-zero elements of β .

There will be a bias-variance tradeoff. As k increases, the bias decreases but the variance increases.

We can approximate the risk with the training error. But the minimization is over all subsets of size k . This minimization is NP-hard. So best subset regression is infeasible. We can approximate best subset regression in two different ways: a greedy approximation or a convex relaxation. The former leads to forward stepwise regression. The latter leads to the lasso.

All these methods involve a tuning parameter which can be chosen by cross-validation.

3 Ridge Regression

In this case we minimize

$$\frac{1}{n} \sum_i (Y_i - X_i^T \beta)^2 + \lambda \|\beta\|^2$$

Forward Stepwise Regression

1. Input k . Let $S = \emptyset$.
2. Let $r_j = n^{-1} \sum_i Y_i X_i(j)$ denote the correlation between Y and the j^{th} feature.
Let $J = \operatorname{argmax}_j |r_j|$. Let $S = S \cup \{J\}$.
3. Compute the regression of Y on $X_S = (X(j) : j \in S)$. Compute the residuals $e = (e_1, \dots, e_n)$ where $e_i = Y_i - \hat{\beta}_S^T X_i$.
4. Compute the correlations r_j between the residuals e and the remaining features.
5. Let $J = \operatorname{argmax}_j |r_j|$. Let $S = S \cup \{J\}$.
6. Repeat steps 3-5 until $|S| = k$.
7. Output S .

Figure 1: Forward Stepwise Regression

where $\lambda \geq 0$. The minimizer is

$$\hat{\beta} = (\hat{\Sigma} + \lambda I)^{-1} \hat{\alpha}.$$

As λ increases, the bias increases and the variance decreases.

Theorem 4 (Hsu, Kakade and Zhang 2014) Suppose that $\|X_i\| \leq r$. Let $\beta^T x$ be the best linear approximation to $m(x)$. Then, with probability at least $1 - 4e^{-t}$,

$$r(\hat{\beta}) - r(\beta) \leq \left(1 + O\left(\frac{1 + \frac{r^2}{\lambda}}{n}\right)\right) \frac{\lambda \|\beta\|^2}{2} + \frac{\sigma^2 \operatorname{tr}(\Sigma)}{n} \frac{\lambda}{2\lambda}.$$

Proposition 5 If $Y = X^T \beta + \epsilon$, $\epsilon \sim N(0, \sigma^2)$ and $\beta \sim N(0, \tau^2 I)$. Then the posterior mean is the ridge regression estimator with $\lambda = \sigma^2 / \tau^2$.

4 Forward Stepwise Regression (Greedy Regression)

Forward stepwise regression is a greedy approximation to best subset regression. In what follows, we will assume that the features have been standardized to have sample mean 0 and sample variance $n^{-1} \sum_i X_i^2(j) = 1$. The algorithm is in Figure 1.

Now we will discuss the theory of forward stepwise regression. Let's start with a functional, noise-free version. We want to greedily approximate a function f using a dictionary of functions $\mathcal{D} = \{\psi_1, \psi_2, \dots\}$. The elements of \mathcal{D} are called atoms. Assume that $\|\psi\| = 1$ for all $\psi \in \mathcal{D}$. Assume that f and the atoms of the dictionary belong to a Hilbert space \mathcal{H} .

1. Input: f .
2. Initialize: $r_0 = f$, $f_0 = 0$, $V = \emptyset$.
3. Repeat: At step N define

$$g_N = \operatorname{argmax}_{\psi \in \mathcal{D}} |\langle r_{N-1}, \psi \rangle|$$

and set $V_N = V_{N-1} \cup \{g_N\}$. Let f_N be the projection of r_{N-1} onto $\operatorname{Span}(V_N)$. Let $r_N = f - f_N$.

Figure 2: The Orthogonal Greedy Algorithm.

Let Σ_N denote all linear combinations of elements of \mathcal{D} with at most N terms. Define the best N -term approximation error

$$\sigma_N(f) = \inf_{|\Lambda| \leq N} \inf_{g \in \operatorname{Span}(\Lambda)} \|f - g\| \quad (3)$$

where Λ denotes a subset of \mathcal{D} and $\operatorname{Span}(\Lambda)$ is the set of linear combinations of functions in Λ .

Suppose first that f is in the span of the dictionary. The function may then have more than one expansion of the form $f = \sum_j \beta_j \psi_j$. We define the norm

$$\|f\|_{\mathcal{L}_p} = \inf \|\beta\|_p$$

where the infimum is over all expansions of f . The functional version of stepwise regression, known as the **Orthogonal Greedy Algorithm** (OGA), is also known as Orthogonal Matching Pursuit. The algorithm is given in Figure 2.

The algorithm produces a series of approximations f_N with corresponding residuals r_N . We have the following two theorems from Barron et al (2008), the first dating back to DeVore and Temlyakov (1996).

Theorem 6 *For all $f \in \mathcal{L}_1$, the residual r_N after N steps of OGA satisfies*

$$\|r_N\| \leq \frac{\|f\|_{\mathcal{L}_1}}{\sqrt{N+1}} \quad (4)$$

for all $N \geq 1$.

Proof. Note that f_N is the best approximation to f from $\operatorname{Span}(V_N)$. On the other hand, the best approximation from the set $\{a g_N : a \in \mathbb{R}\}$ is $\langle f, g_N \rangle g_N$. The error of the former must be

smaller than the error of the latter. In other words, $\|f - f_N\|^2 \leq \|f - f_{N-1} - \langle r_{N-1}, g_N \rangle g_N\|^2$. Thus,

$$\begin{aligned}\|r_N\|^2 &\leq \|r_{N-1} - \langle r_{N-1}, g_N \rangle g_N\|^2 \\ &= \|r_{N-1}\|^2 + |\langle r_{N-1}, g_N \rangle|^2 \underbrace{\|g_N\|^2}_{=1} - 2|\langle r_{N-1}, g_N \rangle|^2 \\ &= \|r_{N-1}\|^2 - |\langle r_{N-1}, g_N \rangle|^2.\end{aligned}\tag{5}$$

Now, $f = f_{N-1} + r_{N-1}$ and $\langle f_{N-1}, r_{N-1} \rangle = 0$. So,

$$\begin{aligned}\|r_{N-1}\|^2 &= \langle r_{N-1}, r_{N-1} \rangle = \langle r_{N-1}, f - f_{N-1} \rangle = \langle r_{N-1}, f \rangle - \underbrace{\langle r_{N-1}, f_{N-1} \rangle}_{=0} \\ &= \langle r_{N-1}, f \rangle = \sum_j \beta_j \langle r_{N-1}, \psi_j \rangle \leq \sup_{\psi \in \mathcal{D}} |\langle r_{N-1}, \psi \rangle| \sum_j |\beta_j| \\ &= \sup_{\psi \in \mathcal{D}} |\langle r_{N-1}, \psi \rangle| \|f\|_{\mathcal{L}_1} = |\langle r_{N-1}, g_N \rangle| \|f\|_{\mathcal{L}_1}.\end{aligned}$$

Continuing from equation (5), we have

$$\begin{aligned}\|r_N\|^2 &\leq \|r_{N-1}\|^2 - |\langle r_{N-1}, g_N \rangle|^2 = \|r_{N-1}\|^2 \left(1 - \frac{\|r_{N-1}\|^2 |\langle r_{N-1}, g_N \rangle|^2}{\|r_{N-1}\|^4}\right) \\ &\leq \|r_{N-1}\|^2 \left(1 - \frac{\|r_{N-1}\|^2 |\langle r_{N-1}, g_N \rangle|^2}{|\langle r_{N-1}, g_N \rangle|^2 \|f\|_{\mathcal{L}_1}^2}\right) = \|r_{N-1}\|^2 \left(1 - \frac{\|r_{N-1}\|^2}{\|f\|_{\mathcal{L}_1}^2}\right).\end{aligned}$$

If $a_0 \geq a_1 \geq a_2 \geq \dots$ are nonnegative numbers such that $a_0 \leq M$ and $a_N \leq a_{N-1}(1 - a_{N-1}/M)$ then it follows from induction that $a_N \leq M/(N+1)$. The result follows by setting $a_N = \|r_N\|^2$ and $M = \|f\|_{\mathcal{L}_1}^2$. \square

If f is not in \mathcal{L}_1 , it is still possible to bound the error as follows.

Theorem 7 For all $f \in \mathcal{H}$ and $h \in \mathcal{L}_1$,

$$\|r_N\|^2 \leq \|f - h\|^2 + \frac{4\|h\|_{\mathcal{L}_1}^2}{N}.\tag{6}$$

Proof. Choose any $h \in \mathcal{L}_1$ and write $h = \sum_j \beta_j \psi_j$ where $\|h\|_{\mathcal{L}_1} = \sum_j |\beta_j|$. Write $f = f_{N-1} + f - f_{N-1} = f_{N-1} + r_{N-1}$ and note that r_{N-1} is orthogonal to f_{N-1} . Hence, $\|r_{N-1}\|^2 =$

$\langle r_{N-1}, f \rangle$ and so

$$\begin{aligned}
\|r_{N-1}\|^2 &= \langle r_{N-1}, f \rangle = \langle r_{N-1}, h + f - h \rangle = \langle r_{N-1}, h \rangle + \langle r_{N-1}, f - h \rangle \\
&\leq \langle r_{N-1}, h \rangle + \|r_{N-1}\| \|f - h\| \\
&= \sum_j \beta_j \langle r_{N-1}, \psi_j \rangle + \|r_{N-1}\| \|f - h\| \\
&\leq \sum_j |\beta_j| |\langle r_{N-1}, \psi_j \rangle| + \|r_{N-1}\| \|f - h\| \\
&\leq \max_j |\langle r_{N-1}, \psi_j \rangle| \sum_j |\beta_j| + \|r_{N-1}\| \|f - h\| \\
&= |\langle r_{N-1}, g_k \rangle| \|h\|_{\mathcal{L}_1} + \|r_{N-1}\| \|f - h\| \\
&\leq |\langle r_{N-1}, g_k \rangle| \|h\|_{\mathcal{L}_1} + \frac{1}{2} (\|r_{N-1}\|^2 + \|f - h\|^2).
\end{aligned}$$

Hence,

$$|\langle r_{N-1}, g_k \rangle|^2 \geq \frac{(\|r_{N-1}\|^2 - \|f - h\|^2)^2}{4\|h\|_{\mathcal{L}_1}^2}.$$

Thus,

$$a_N \leq a_{N-1} \left(1 - \frac{a_{N-1}}{4\|h\|_{\mathcal{L}_1}^2} \right)$$

where $a_N = \|r_N\|^2 - \|f - h\|^2$. By induction, the last displayed inequality implies that $a_N \leq 4\|h\|_{\mathcal{L}_1}^2/k$ and the result follows. \square

Corollary 8 For each N ,

$$\|r_N\|^2 \leq \sigma_N^2 + \frac{4\theta_N^2}{N}$$

where θ_N is the \mathcal{L}_1 norm of the best N -atom approximation.

In Figure 3 we re-express forward stepwise regression in a form closer to the notation we have been using. In this version, we have a finite dictionary \mathcal{D}_n and a data vector $Y = (Y_1, \dots, Y_n)^T$ and we use the empirical norm defined by

$$\|h\|_n = \sqrt{\frac{1}{n} \sum_{i=1}^n h^2(X_i)}.$$

We assume that the dictionary is normalized in this empirical norm.

By combining the previous results with concentration of measure arguments (see appendix for details) we get the following result, due to Barron, Cohen, Dahmen and DeVore (2008).

1. Input: $Y \in \mathbb{R}^n$.
2. Initialize: $r_0 = Y$, $\hat{f}_0 = 0$, $V = \emptyset$.
3. Repeat: At step N define

$$g_N = \operatorname{argmax}_{\psi \in \mathcal{D}} |\langle r_{N-1}, \psi \rangle_n|$$

where $\langle a, b \rangle_n = n^{-1} \sum_{i=1}^n a_i b_i$. Set $V_N = V_{N-1} \cup \{g_N\}$. Let f_N be the projection of r_{N-1} onto $\operatorname{Span}(V_N)$. Let $r_N = Y - f_N$.

Figure 3: The Greedy (Forward Stepwise) Regression Algorithm: Dictionary Version

Theorem 9 *Let $h_n = \operatorname{argmin}_{h \in \mathcal{F}_N} \|f_0 - h\|^2$. Suppose that $\limsup_{n \rightarrow \infty} \|h_n\|_{\mathcal{L}_{1,n}} < \infty$. Let $N \sim \sqrt{n}$. Then, for every $\gamma > 0$, there exist $C > 0$ such that*

$$\|f - \hat{f}_N\|^2 \leq 4\sigma_N^2 + \frac{C \log n}{n^{1/2}}$$

except on a set of probability $n^{-\gamma}$.

Let us compare this with the lasso which we will discuss next. Let $f_L = \sum_j \beta_j \psi_j$ minimize $\|f - f_L\|^2$ subject to $\|\beta\|_1 \leq L$. Then, we will see that

$$\|f - \hat{f}_L\|^2 \leq \|f - f_L\|^2 + O_P \left(\frac{\log n}{n} \right)^{1/2}$$

which is the same rate.

The rate $n^{-1/2}$ is in fact optimal. It might be surprising that the rate is independent of the dimension. Why do you think this is the case?

4.1 The Lasso

The lasso approximates best subset regression by using a convex relaxation. In particular, the norm $\|\beta\|_0$ is replaced with $\|\beta\|_1 = \sum_j |\beta_j|$.

The lasso estimator $\hat{\beta}$ is defined as the minimizer of

$$\sum_i (Y_i - \beta^T X_i)^2 + \lambda \|\beta\|_1.$$

This is a convex problem so the estimator can be found efficiently. The estimator is sparse: for large enough λ , many of the components of $\hat{\beta}$ are 0. This is proved in the course on convex optimization. Now we discuss some theoretical properties of the lasso.¹

The following result was proved in Zhao and Yu (2006), Meinshausen and Bühlmann (2005) and Wainwright (2006). The version we state is from Wainwright (2006). Let $\beta = (\beta_1, \dots, \beta_s, 0, \dots, 0)$ and decompose the design matrix as $\mathbb{X} = (\mathbb{X}_S \ \mathbb{X}_{S^c})$ where $S = \{1, \dots, s\}$. Let $\beta_S = (\beta_1, \dots, \beta_s)$.

Theorem 10 (Sparsistency) *Suppose that:*

1. *The true model is linear.*
2. *The design matrix satisfies*

$$\|\mathbb{X}_{S^c}\mathbb{X}_S(\mathbb{X}_S^T\mathbb{X}_S)^{-1}\|_\infty \leq 1 - \epsilon \quad \text{for some } 0 < \epsilon \leq 1. \quad (7)$$

3. $\phi_n(d_n) > 0$.
4. *The ϵ_i are Normal.*

5. λ_n *satisfies*

$$\frac{n\lambda_n^2}{\log(d_n - s_n)} \rightarrow \infty$$

and

$$\frac{1}{\min_{1 \leq j \leq s_n} |\beta_j|} \left(\sqrt{\frac{\log s_n}{n}} + \lambda_n \left\| \left(\frac{1}{n} \mathbb{X}^T \mathbb{X} \right)^{-1} \right\|_\infty \right) \rightarrow 0. \quad (8)$$

Then the lasso is sparsistent, meaning that $P(\text{support}(\hat{\beta}) = \text{support}(\beta)) \rightarrow 1$ where $\text{support}(\beta) = \{j : \beta(j) \neq 0\}$.

The conditions of this theorem are very strong. They are not checkable and they are unlikely to ever be true in practice.

Theorem 11 (Consistency: Meinshausen and Yu 2006) *Assume that*

1. *The true regression function is linear.*
2. *The columns of \mathbb{X} have norm n and the covariates are bounded.*

¹The norm $\|\beta\|_1$ can be thought of as a measure of sparsity. For example, the vectors $x = (1/\sqrt{d}, \dots, 1/\sqrt{d})$ and $y = (1, 0, \dots, 1)$ have the same L_2 norm. But $\|y\|_1 = 1 < \|x\|_1 = \sqrt{d}$.

- 3. $\mathbb{E}(\exp|\epsilon_i|) < \infty$ and $\mathbb{E}(\epsilon_i^2) = \sigma^2 < \infty$.
- 4. $\mathbb{E}(Y_i^2) \leq \sigma_y^2 < \infty$.
- 5. $0 < \phi_n(k_n) \leq \Phi_n(k_n) < \infty$ for $k_n = \min\{n, d_n\}$.
- 6. $\liminf_{n \rightarrow \infty} \phi_n(s_n \log n) > 0$ where $s_n = \|\beta_n\|_0$.

Then

$$\|\beta_n - \hat{\beta}_n\|^2 = O_P \left(\frac{\log n}{n} \frac{s_n \log n}{\phi_n^2(s_n \log n)} \right) + O \left(\frac{1}{\log n} \right) \quad (9)$$

If

$$s_n \log d_n \left(\frac{\log n}{n} \right) \rightarrow 0 \quad (10)$$

and

$$\lambda_n = \sqrt{\frac{\sigma_y^2 \Phi_n(\min n, d_n) n^2}{s_n \log n}} \quad (11)$$

then $\|\hat{\beta}_n - \beta_n\|^2 \xrightarrow{P} 0$.

Once again, the conditions of this theorem are very strong. They are not checkable and they are unlikely to ever be true in practice.

The next theorem is the most important one. It does not require unrealistic conditions. We state the theorem for bounded covariates. A more general version appears in Greenshtein and Ritov (2004).

Theorem 12 Let $Z = (Y, X)$. Assume that $|Y| \leq B$ and $\max_j |X(j)| \leq B$. Let

$$\beta_* = \underset{\|\beta\|_1 \leq L}{\operatorname{argmin}} r(\beta)$$

where $r(\beta) = \mathbb{E}(Y - \beta^T X)^2$. Thus, $x^T \beta_*$ is the best, sparse linear predictor (in the L_1 sense). Let $\hat{\beta}$ be the lasso estimator:

$$\hat{\beta} = \underset{\|\beta\|_1 \leq L}{\operatorname{argmin}} \hat{r}(\beta)$$

where $\hat{r}(\beta) = n^{-1} \sum_{i=1}^n (Y_i - X_i^T \beta)^2$. With probability at least $1 - \delta$,

$$r(\hat{\beta}) \leq r(\beta_*) + \sqrt{\frac{16(L+1)^4 B^2}{n} \log \left(\frac{\sqrt{2} d}{\sqrt{\delta}} \right)}.$$

Proof. Let $Z = (Y, X)$ and $Z_i = (Y_i, X_i)$. Define $\gamma \equiv \gamma(\beta) = (-1, \beta)$. Then

$$r(\beta) = \mathbb{E}(Y - \beta^T X)^2 = \gamma^T \Lambda \gamma$$

where $\Lambda = \mathbb{E}[ZZ^T]$. Note that $\|\gamma\|_1 = \|\beta\|_1 + 1$. Let $\mathcal{B} = \{\beta : \|\beta\|_1 \leq L\}$. The training error is

$$\hat{r}(\beta) = \frac{1}{n} \sum_{i=1}^n (Y_i - X_i^T \beta)^2 = \gamma^T \hat{\Lambda} \gamma$$

where $\hat{\Lambda} = \frac{1}{n} \sum_{i=1}^n Z_i Z_i^T$. For any $\beta \in \mathcal{B}$,

$$\begin{aligned} |\hat{r}(\beta) - r(\beta)| &= |\gamma^T (\hat{\Lambda} - \Lambda) \gamma| \\ &\leq \sum_{j,k} |\gamma(j)| |\gamma(k)| |\hat{\Lambda}(j, k) - \Lambda(j, k)| \leq \|\gamma\|_1^2 \delta_n \\ &\leq (L+1)^2 \Delta_n \end{aligned}$$

where

$$\Delta_n = \max_{j,k} |\hat{\Lambda}(j, k) - \Lambda(j, k)|.$$

So,

$$r(\hat{\beta}) \leq \hat{r}(\hat{\beta}) + (L+1)^2 \Delta_n \leq \hat{r}(\beta_*) + (L+1)^2 \Delta_n \leq r(\beta_*) + 2(L+1)^2 \Delta_n.$$

Note that $|Z(j)Z(k)| \leq B^2 < \infty$. By Hoeffding's inequality,

$$\mathbb{P}(\Delta_n(j, k) \geq \epsilon) \leq 2e^{-n\epsilon^2/(2B^2)}$$

and so, by the union bound,

$$\mathbb{P}(\Delta_n \geq \epsilon) \leq 2d^2 e^{-n\epsilon^2/(2B^2)} = \delta$$

if we choose $\epsilon = \sqrt{(4B^2/n) \log \left(\frac{\sqrt{2}d}{\sqrt{\delta}} \right)}$. Hence,

$$r(\hat{\beta}) \leq r(\beta_*) + \sqrt{\frac{16(L+1)^4 B^2}{n} \log \left(\frac{\sqrt{2}d}{\sqrt{\delta}} \right)}.$$

with probability at least $1 - \delta$. \square

Problems With Sparsity. Sparse estimators are convenient and popular but they can some problems. Say that $\hat{\beta}$ is **weakly sparsistent** if, for every β ,

$$P_\beta(I(\hat{\beta}_j = 1) \leq I(\beta_j = 1) \text{ for all } j) \rightarrow 1 \quad (12)$$

as $n \rightarrow \infty$. In particular, if $\hat{\beta}_n$ is sparsistent, then it is weakly sparsistent. Suppose that d is fixed. Then the least squares estimator $\hat{\beta}_n$ is minimax and satisfies

$$\sup_{\beta} E_\beta(n\|\hat{\beta}_n - \beta\|^2) = O(1). \quad (13)$$

But sparsistent estimators have much larger risk:

Theorem 13 (Leeb and Pötscher (2007)) Suppose that the following conditions hold:

1. d is fixed.
2. The covariates are nonstochastic and $n^{-1}\mathbb{X}^T\mathbb{X} \rightarrow Q$ for some positive definite matrix Q .
3. The errors ϵ_i are independent with mean 0, finite variance σ^2 and have a density f satisfying

$$0 < \int \left(\frac{f'(x)}{f(x)} \right)^2 f(x) dx < \infty.$$

If $\hat{\beta}$ is weakly sparsistent then

$$\sup_{\beta} E_{\beta}(n\|\hat{\beta}_n - \beta\|^2) \rightarrow \infty. \quad (14)$$

More generally, if ℓ is any nonnegative loss function then

$$\sup_{\beta} E_{\beta}(\ell(n^{1/2}(\hat{\beta}_n - \beta))) \rightarrow \sup_s \ell(s). \quad (15)$$

Proof. Choose any $s \in \mathbb{R}^d$ and let $\beta_n = -s/\sqrt{n}$. Then,

$$\begin{aligned} \sup_{\beta} E_{\beta}(\ell(n^{1/2}(\hat{\beta} - \beta))) &\geq E_{\beta_n}(\ell(n^{1/2}(\hat{\beta} - \beta))) \geq E_{\beta_n}(\ell(n^{1/2}(\hat{\beta} - \beta))I(\hat{\beta} = 0)) \\ &= \ell(-\sqrt{n}\beta_n)P_{\beta_n}(\hat{\beta} = 0) = \ell(s)P_{\beta_n}(\hat{\beta} = 0). \end{aligned}$$

Now, $P_0(\hat{\beta} = 0) \rightarrow 1$ by assumption. It can be shown that we also have $P_{\beta_n}(\hat{\beta} = 0) \rightarrow 1$.² Hence, with probability tending to 1,

$$\sup_{\beta} E_{\beta}(\ell(n^{1/2}(\hat{\beta} - \beta))) \geq \ell(s).$$

Since s was arbitrary the result follows. \square

It follows that, if R_n denotes the minimax risk then

$$\sup_{\beta} \frac{R(\hat{\beta}_n)}{R_n} \rightarrow \infty.$$

The implication is that when d is much smaller than n , sparse estimators have poor behavior. However, when d_n is increasing and $d_n > n$, the least squares estimator no longer satisfies (13). Thus we can no longer say that some other estimator outperforms the sparse estimator. In summary, sparse estimators are well-suited for high-dimensional problems but not for low dimensional problems.

²This follows from a property called contiguity.

5 Cross Validation

The following result is from Gyorfi, Kohler, Krzyzak and Walk (2002). Let $\mathcal{M} = \{m_h\}$ be a finite class of regression estimators indexed by a parameter h . Let $m_{\hat{h}}$ minimize $\int |m_h(x) - m(x)|^2 dP(x)$ over \mathcal{M} . We want to show that cross-validation (data-splitting) leads to an estimator with risk nearly as good as $m_{\hat{h}}$.

Split the data into training $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ and test $\mathcal{D}' = \{(X'_1, Y'_1), \dots, (X'_n, Y'_n)\}$. Let m_H minimize $n^{-1} \sum_{i \in \mathcal{D}'} |Y_i - m_h(X_i)|^2$. Assume that the data Y_i and the estimators are bounded by L .

Theorem 14 *For any $\delta > 0$,*

$$\mathbb{E} \int |m_H(x) - m(x)|^2 dP(x) \leq (1 + \delta) \mathbb{E} \int |m_{\hat{h}}(x) - m(x)|^2 dP(x) + \frac{C(1 + \log |M|)}{n}$$

where $c = L^2(16/\delta + 35 + 19\delta)$.

Proof. Then

$$\begin{aligned} \mathbb{E} \left(\int |m_H - m|^2 dP(x) | \mathcal{D} \right) &= \mathbb{E} \left(\int |Y - m_H|^2 dP(x) | \mathcal{D} \right) - \mathbb{E} |Y - m(X)|^2 \\ &= T_1 + T_2 \end{aligned}$$

where

$$T_1 = \mathbb{E} \left(\int |Y - m_H|^2 dP(x) | \mathcal{D} \right) - \mathbb{E} |Y - m(X)|^2 - T_2$$

and

$$T_2 = (1 + \delta) \frac{1}{n} \sum_{\mathcal{D}'} (|Y_i - m_H(X_i)|^2 - |Y_i - m(X_i)|^2) \leq (1 + \delta) \frac{1}{n} \sum_{\mathcal{D}'} (|Y_i - m_{\hat{h}}(X_i)|^2 - |Y_i - m(X_i)|^2)$$

and so

$$\begin{aligned} \mathbb{E}[T_2 | \mathcal{D}] &\leq (1 + \delta) (\mathbb{E}(|Y - m_{\hat{h}}(X)|^2 | \mathcal{D}) - \mathbb{E}|Y - m(X)|^2) \\ &= (1 + \delta) \int |m_{\hat{h}}(x) - m(x)|^2 dP(x). \end{aligned}$$

The second part of the proof involves some tedious calculations. We will bound $P(T_1 \geq s | \mathcal{D})$. The event $T_1 \geq s$ is the same as

$$\begin{aligned} (1 + \delta) \left(\mathbb{E}(|m_H(X) - Y|^2 | \mathcal{D}) - \mathbb{E}|m(X) - Y|^2 - \frac{1}{n} \sum_{\mathcal{D}'} (|Y_i - m_H(X_i)|^2 - |Y_i - m(X_i)|^2) \right) &\geq \\ s + \delta (\mathbb{E}|m_H(X) - Y|^2 - \mathbb{E}|m(X) - Y|^2) . \end{aligned}$$

This has probability at most $|\mathcal{M}|$ times the probability that

$$(1 + \delta) \left(\mathbb{E}(|m_h(X) - Y|^2 | \mathcal{D}) - \mathbb{E}|m(X) - Y|^2 - \frac{1}{n} \sum_{\mathcal{D}'} (|Y_i - m_H(X_i)|^2 - |Y_i - m(X_i)|^2) \right) \geq s + \delta (\mathbb{E}|m_h(X) - Y|^2 - \mathbb{E}|m(X) - Y|^2)$$

for some h , that is

$$\mathbb{E}[Z | \mathcal{D}] - \frac{1}{n} \sum_i Z_i \geq \frac{s + \delta \mathbb{E}[Z | \mathcal{D}]}{1 + \delta}$$

for some h , where $Z = |m_h(X) - Y|^2 - |m(X) - Y|^2$. Now

$$\sigma^2 = \text{Var}(Z | \mathcal{D}) \leq \mathbb{E}[Z^2 | \mathcal{D}] \leq 16L^2 \int |m_h(x) - m(x)|^2 dP(x) = 16L^2 \mathbb{E}[Z | \mathcal{D}].$$

Using this, and Bernstein's inequality,

$$\begin{aligned} & P \left(\mathbb{E}[Z | \mathcal{D}] - \bar{Z} \geq \frac{s + \delta \mathbb{E}[Z | \mathcal{D}]}{1 + \delta} | \mathcal{D} \right) \\ & \leq P \left(\mathbb{E}[Z | \mathcal{D}] - \bar{Z} \geq \frac{s + \delta \sigma^2 / (16L^2)}{1 + \delta} | \mathcal{D} \right) \\ & \leq e^{-nA/B} \end{aligned}$$

where

$$A = \frac{1}{(1 + \delta)^2} \left(s + \frac{\delta \sigma^2}{16L^2} \right)$$

and

$$B = 2\sigma^2 + \frac{2}{3} \frac{8L^2}{1 + \delta} \left(s + \frac{\delta \sigma^2}{16L^2} \right).$$

Now $A/B \geq s/c$ for $c = L^2(16/\delta + 35 + 19\delta)$. So

$$P(T_1 \geq s | \mathcal{D}) \leq |\mathcal{M}| e^{-ns/c}.$$

Finally

$$\mathbb{E}[T_1 | \mathcal{D}] \leq u + \int_u^\infty P(T_1 > s | \mathcal{D}) \leq u + \frac{c|\mathcal{M}|}{n} e^{-nu/c}.$$

The result follows by setting $u = c \log |\mathcal{M}|/n$. \square

6 Inference?

Is it possible to do inference after model selection? Do we need to? I'll discuss this in class.

References

- Buja, Berk, Brown, George, Pitkin, Traskin, Zhao and Zhang (2015). Models as Approximations — A Conspiracy of Random Regressors and Model Deviations Against Classical Inference in Regression. *Statistical Science*.
- Hsu, Kakade and Zhang (2014). Random design analysis and ridge regression. arXiv:1106.2363.
- Gyorfi, Kohler, Krzyzak and Walk. (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer.

Appendix: L_2 Boosting

Define estimators $\widehat{m}_n^{(0)}, \dots, \widehat{m}_n^{(k)}, \dots$, as follows. Let $\widehat{m}^{(0)}(x) = 0$ and then iterate the following steps:

1. Compute the residuals $U_i = Y_i - \widehat{m}^{(k)}(X_i)$.
2. Regress the residuals on the Y_i 's: $\widehat{\beta}_j = \sum_i U_i X_{ij} / \sum_i X_{ij}^2$, $j = 1, \dots, d$.
3. Find $J = \operatorname{argmin}_j RSS_j$ where $RSS_j = \sum_i (U_i - \widehat{\beta}_J X_{iJ})^2$.
4. Set $\widehat{m}^{(k+1)}(x) = \widehat{m}^{(k)}(x) + \widehat{\beta}_J x_J$.

The version above is called **L_2 boosting** or **matching pursuit**. A variation is to set $\widehat{m}^{(k+1)}(x) = \widehat{m}^{(k)}(x) + \nu \widehat{\beta}_J x_J$ where $0 < \nu \leq 1$. Another variation is to set $\widehat{m}^{(k+1)}(x) = \widehat{m}^{(k)}(x) + \nu \operatorname{sign}(\widehat{\beta}_J) x_J$ which is called **forward stagewise regression**. Yet another variation is to set $\widehat{m}^{(k)}$ to be the linear regression estimator based on all variables selected up to that point. This is **forward stepwise regression** or **orthogonal matching pursuit**.

Theorem 15 *The matching pursuit estimator is linear. In particular,*

$$\widehat{Y}^{(k)} = B_k Y \tag{16}$$

where $\widehat{Y}^{(k)} = (\widehat{m}^{(k)}(X_1), \dots, \widehat{m}^{(k)}(X_n))^T$,

$$B_k = I - (I - H_k)(I - H_{k-1}) \cdots (I - H_1), \tag{17}$$

and

$$H_j = \frac{\mathbb{X}_j \mathbb{X}_j^T}{\|\mathbb{X}_j\|^2}. \tag{18}$$

Theorem 16 (Bühlmann 2005) Let $m_n(x) = \sum_{j=1}^{d_n} \beta_{j,n} x_j$ be the best linear approximation based on d_n terms. Suppose that:

(A1 Growth) $d_n \leq C_0 e^{C_1 n^{1-\xi}}$ for some $C_0, C_1 > 0$ and some $0 < \xi \leq 1$.

(A2 Sparsity) $\sup_n \sum_{j=1}^{d_n} |\beta_{j,n}| < \infty$.

(A3 Bounded Covariates) $\sup_n \max_{1 \leq j \leq d_n} \max_i |X_{ij}| < \infty$ with probability 1.

(A4 Moments) $\mathbb{E}|\epsilon|^s < \infty$ for some $s > 4/\xi$.

Then there exists $k_n \rightarrow \infty$ such that

$$\mathbb{E}_X |\hat{m}_n(X) - m_n(x)|^2 \rightarrow 0 \quad (19)$$

as $n \rightarrow 0$.

We won't prove the theorem but we will outline the idea. Let \mathcal{H} be a Hilbert space with inner product $\langle f, g \rangle = \int f(x)g(x)dP(x)$. Let \mathcal{D} be a dictionary, that is a set of functions, each of unit norm, that span \mathcal{H} . Define a functional version of matching pursuit, known as the **weak greedy algorithm**, as follows. Let $R_0(f) = f$, $F_0 = 0$. At step k , find $g_k \in \mathcal{D}$ so that

$$|\langle R_{k-1}(f), g_k \rangle| \geq t_k \sup_{h \in \mathcal{D}} |\langle R_{k-1}(f), h \rangle|$$

for some $0 < t_k \leq 1$. In the weak greedy algorithm we take $F_k = F_{k-1} + \langle f, g_k \rangle g_k$. In the weak orthogonal greedy algorithm we take F_k to be the projection of $R_{k-1}(f)$ onto $\{g_1, \dots, g_k\}$. Finally set $R_k(f) = f - F_k$.

Theorem 17 (Temlyakov 2000) Let $f(x) = \sum_j \beta_j g_j(x)$ where $g_j \in \mathcal{D}$ and $\sum_{j=1}^{\infty} |\beta_j| \leq B < \infty$. Then, for the weak orthogonal greedy algorithm

$$\|R_k(f)\| \leq \frac{B}{\left(1 + \sum_{j=1}^k t_j^2\right)^{1/2}} \quad (20)$$

and for the weak greedy algorithm

$$\|R_k(f)\| \leq \frac{B}{\left(1 + \sum_{j=1}^k t_j^2\right)^{t_k/(2(2+t_k))}}. \quad (21)$$

L_2 boosting essentially replaces $\langle f, X_j \rangle$ with $\langle Y, X_j \rangle_n = n^{-1} \sum_i Y_i X_{ij}$. Now $\langle Y, X_j \rangle_n$ has mean $\langle f, X_j \rangle$. The main burden of the proof is to show that $\langle Y, X_j \rangle_n$ is close to $\langle f, X_j \rangle$ with

high probability and then apply Temlyakov's result. For this we use Bernstein's inequality. Recall that if $|Z_j|$ are bounded by M and Z_j has variance σ^2 then

$$\mathbb{P}(|\bar{Z} - \mathbb{E}(Z_j)| > \epsilon) \leq 2 \exp \left\{ -\frac{1}{2} \frac{n\epsilon^2}{\sigma^2 + M\epsilon/3} \right\}. \quad (22)$$

Hence, the probability that any empirical inner products differ from their functional counterparts is no more than

$$d_n^2 \exp \left\{ -\frac{1}{2} \frac{n\epsilon^2}{\sigma^2 + M\epsilon/3} \right\} \rightarrow 0 \quad (23)$$

because of the growth condition.

Appendix: Proof of Theorem 9

The \mathcal{L}_1 norm depends on n and so we denote this by $\|h\|_{\mathcal{L}_{1,n}}$. For technical reasons, we assume that $\|f\|_\infty \leq B$, that \hat{f}_n is truncated to be no more than B and that $\|\psi\|_\infty \leq B$ for all $\psi \in \mathcal{D}_n$.

Theorem 18 *Suppose that $p_n \equiv |\mathcal{D}|_n \leq n^c$ for some $c \geq 0$. Let \hat{f}_N be the output of the stepwise regression algorithm after N steps. Let $f(x) = \mathbb{E}(Y|X = x)$ denote the true regression function. Then, for every $h \in \mathcal{D}_n$,*

$$\mathbb{P} \left(\|f - \hat{f}_N\|^2 > 4\|f - h\|^2 + \frac{8\|h\|_{\mathcal{L}_{1,n}}^2}{N} + \frac{CN \log n}{n} \right) < \frac{1}{n^\gamma}$$

for some positive constants γ and C .

Before proving this theorem, we need some preliminary results. For any $\Lambda \subset \mathcal{D}$, let $S_\Lambda = \text{Span}(\Lambda)$. Define

$$\mathcal{F}_N = \bigcup \left\{ S_\Lambda : |\Lambda| \leq N \right\}.$$

Recall that, if \mathcal{F} is a set of functions then $N_p(\epsilon, \mathcal{F}, \nu)$ is the L_p covering entropy with respect to the probability measure ν and $N_p(\epsilon, \mathcal{F})$ is the supremum of $N_p(\epsilon, \mathcal{F}, \nu)$ over all probability measures ν .

Lemma 19 *For every $t > 0$, and every $\Lambda \subset \mathcal{D}_n$,*

$$N_1(t, S_\Lambda) \leq 3 \left(\frac{2eB}{t} \log \left(\frac{3eB}{t} \right) \right)^{|\Lambda|+1}, \quad N_2(t, S_\Lambda) \leq 3 \left(\frac{2eB^2}{t^2} \log \left(\frac{3eB^2}{t^2} \right) \right)^{|\Lambda|+1}.$$

Also,

$$N_1(t, \mathcal{F}_N) \leq 12p^N \left(\frac{2eB}{t} \log \left(\frac{3eB}{t} \right) \right)^{N+1}, \quad N_2(t, \mathcal{F}_N) \leq 12p^N \left(\frac{2eB^2}{t^2} \log \left(\frac{3eB^2}{t^2} \right) \right)^{N+1}.$$

Proof. The first two equation follow from standard covering arguments. The second two equations follow from the fact that the number of subsets of Λ of size at most N is

$$\sum_{j=1}^N \binom{p}{j} \leq \sum_{j=1}^N \left(\frac{ep}{j} \right)^j \leq N \left(\frac{ep}{N} \right)^N \leq p^N \max_N N \left(\frac{p}{N} \right)^N \leq 4p^N.$$

□

The following lemma is from Chapter 11 of Gyorfi et al. The proof is long and technical and we omit it.

Lemma 20 Suppose that $|Y| \leq B$, where $B \geq 1$, and \mathcal{F} is a set of real-valued functions such that $\|f\|_\infty \leq B$ for all $f \in \mathcal{F}$. Let $f_0(x) = \mathbb{E}(Y|X = x)$ and $\|g\|^2 = \int g^2(x)dP(x)$. Then, for every $\alpha, \beta > 0$ and $\epsilon \in (0, 1/2]$,

$$\begin{aligned} & \mathbb{P} \left((1 - \epsilon) \|f - f_0\|^2 \geq \|Y - f\|_n^2 - \|Y - f_0\|_n^2 + \epsilon(\alpha + \beta) \quad \text{for some } f \in \mathcal{F} \right) \\ & \leq 14N_1 \left(\frac{\beta\epsilon}{20B}, \mathcal{F} \right) \exp \left\{ -\frac{\epsilon^2(1 - \epsilon)\alpha n}{214(1 + \epsilon)B^4} \right\}. \end{aligned}$$

Proof of Theorem 18. For any $h \in \mathcal{F}_n$ we have

$$\begin{aligned} \|\widehat{f} - f_0\|_n^2 &= \underbrace{\|\widehat{f} - f_0\|^2 - 2 \left(\|Y - \widehat{f}\|_n^2 - \|Y - f_0\|_n^2 \right)}_{A_1} \\ &\quad + \underbrace{2 \left(\|Y - \widehat{f}\|_n^2 - \|Y - h\|_n^2 \right)}_{A_2} + \underbrace{2 \left(\|Y - h\|_n^2 - \|Y - f_0\|_n^2 \right)}_{A_3}. \end{aligned}$$

Apply Lemma 20 with $\epsilon = 1/2$ together with Lemma 19 to conclude that, for $C_0 > 0$ large enough,

$$\mathbb{P} \left(A_1 > \frac{C_0 N \log n}{n} \quad \text{for some } f \right) < \frac{1}{n^\gamma}.$$

To bound A_2 , apply Theorem 7 with norm $\|\cdot\|_n$ and with Y replacing f . Then,

$$\|Y - \widehat{f}\|_n^2 \leq \|Y - h\|_n^2 + \frac{4\|h\|_{1,n}^2}{k}$$

and hence $A_2 \leq \frac{8\|h\|_{1,n}^2}{k}$. Next, we have that

$$\mathbb{E}(A_3) = \|f_0 - h\|^2$$

and for large enough C_1 ,

$$\mathbb{P}\left(A_3 > \|f_0 - h\|^2 + \frac{C_1 N \log n}{n} \quad \text{for some } f\right) < \frac{1}{n^\gamma}.$$

□

A Closer Look at Sparse Regression

Ryan Tibshirani

(ammended by Larry Wasserman)

1 Introduction

In these notes we take a closer look at sparse linear regression. Throughout, we make the very strong assumption that $Y_i = \beta^T X_i + \epsilon_i$ where $\mathbb{E}[\epsilon_i|X_i] = 0$ and $\text{Var}(\epsilon_i|X_i) = \sigma^2$. These assumptions are highly unrealistic but they permit a more detailed analysis. There are several books on high-dimensional estimation: [Hastie, Tibshirani & Wainwright \(2015\)](#), [Buhlmann & van de Geer \(2011\)](#), [Wainwright \(2017\)](#).

2 Best subset selection, ridge regression, and the lasso

2.1 Three norms: ℓ_0 , ℓ_1 , ℓ_2

In terms of regularization, we typically choose the constraint set C to be a sublevel set of a norm (or seminorm), and equivalently, the penalty function $P(\cdot)$ to be a multiple of a norm (or seminorm)

Let's consider three canonical choices: the ℓ_0 , ℓ_1 , and ℓ_2 norms:

$$\|\beta\|_0 = \sum_{j=1}^p \mathbb{1}\{\beta_j \neq 0\}, \quad \|\beta\|_1 = \sum_{j=1}^p |\beta_j|, \quad \|\beta\|_2 = \left(\sum_{j=1}^p \beta_j^2 \right)^{1/2}.$$

(Truthfully, calling it “the ℓ_0 norm” is a misnomer, since it is not a norm: it does not satisfy positive homogeneity, i.e., $\|a\beta\|_0 \neq a\|\beta\|_0$ whenever $a \neq 0, 1$.)

In constrained form, this gives rise to the problems:

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to } \|\beta\|_0 \leq k \quad (\text{Best subset selection}) \quad (1)$$

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to } \|\beta\|_1 \leq t \quad (\text{Lasso regression}) \quad (2)$$

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to } \|\beta\|_2^2 \leq t \quad (\text{Ridge regression}) \quad (3)$$

where $k, t \geq 0$ are tuning parameters. Note that it makes sense to restrict k to be an integer; in best subset selection, we are quite literally finding the best subset of variables of size k , in terms of the achieved training error

Though it is likely the case that these ideas were around earlier in other contexts, in statistics we typically subset selection to [Beale et al. \(1967\)](#), [Hocking & Leslie \(1967\)](#), ridge regression to [Hoerl & Kennard \(1970\)](#), and the lasso to [Tibshirani \(1996\)](#), [Chen et al. \(1998\)](#)

In penalized form, the use of ℓ_0, ℓ_1, ℓ_2 norms gives rise to the problems:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_0 \quad (\text{Best subset selection}) \quad (4)$$

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \quad (\text{Lasso regression}) \quad (5)$$

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 \quad (\text{Ridge regression}) \quad (6)$$

with $\lambda \geq 0$ the tuning parameter. In fact, problems (2), (5) are equivalent. By this, we mean that for any $t \geq 0$ and solution $\hat{\beta}$ in (2), there is a value of $\lambda \geq 0$ such that $\hat{\beta}$ also solves (5), and vice versa. The same equivalence holds for (3), (6). (The factors of 1/2 multiplying the squared loss above are inconsequential, and just for convenience)

It means, roughly speaking, that computing solutions of (2) over a sequence of t values and performing cross-validation (to select an estimate) should be basically the same as computing solutions of (5) over some sequence of λ values and performing cross-validation (to select an estimate). Strictly speaking, this isn't quite true, because the precise correspondence between equivalent t, λ depends on the data X, y

Notably, problems (1), (4) are *not equivalent*. For every value of $\lambda \geq 0$ and solution $\hat{\beta}$ in (4), there is a value of $t \geq 0$ such that $\hat{\beta}$ also solves (1), but the converse is not true

2.2 A Toy Example

It is helpful to first consider a toy example. Suppose that $Y \sim N(\mu, 1)$. Let's consider the three different estimators we get using the following three different loss functions:

$$\frac{1}{2}(Y - \mu)^2 + \lambda \|\mu\|_0, \quad \frac{1}{2}(Y - \mu)^2 + \lambda |\mu|, \quad \frac{1}{2}(Y - \mu)^2 + \lambda \mu^2.$$

You should verify that the solutions are

$$\hat{\mu} = H(Y; \sqrt{2\lambda}), \quad \hat{\mu} = S(Y; \lambda), \quad \hat{\mu} = \frac{Y}{1 + 2\lambda}$$

where $H(y; a) = yI(|y| > a)$ is the hard-thresholding operator, and

$$S(y; a) = \begin{cases} y - a & \text{if } y > a \\ 0 & \text{if } -a \leq y \leq a \\ y + a & \text{if } y < a. \end{cases}$$

Hard thresholding creates a “zone of sparsity” but it is discontinuous. Soft thresholding also creates a “zone of sparsity” but it is continuous. The L_2 loss creates a nice smooth estimator but it is never sparse. (You can verify the solution to the L_1 problem using sub-differentials if you know convex analysis, or by doing three cases separately: $\mu > 0, \mu = 0, \mu < 0.$)

2.3 Sparsity

The best subset selection and the lasso estimators have a special, useful property: their solutions are *sparse*, i.e., at a solution $\hat{\beta}$ we will have $\hat{\beta}_j = 0$ for many components $j \in \{1, \dots, p\}$. In problem (1), this is obviously true, where $k \geq 0$ controls the sparsity level. In problem (2), it is less obviously true, but we get a higher degree of sparsity the smaller the value of $t \geq 0$. In the penalized forms, (4), (5), we get more sparsity the larger the value of $\lambda \geq 0$

This is not true of ridge regression, i.e., the solution of (3) or (6) generically has all nonzero components, no matter the value of t or λ . Note that sparsity is desirable, for two reasons: (i) it corresponds to performing variable selection in the constructed linear model, and (ii) it provides a level of interpretability (beyond sheer accuracy)

That the ℓ_0 norm induces sparsity is obvious. But, why does the ℓ_1 norm induce sparsity and not the ℓ_2 norm? There are different ways to look at it; let's stick with intuition from the constrained problem forms (2), (5). Figure 1 shows the “classic” picture, contrasting the way the contours of the squared error loss hit the two constraint sets, the ℓ_1 and ℓ_2 balls. As the ℓ_1 ball has sharp corners (aligned with the coordinate axes), we get sparse solutions

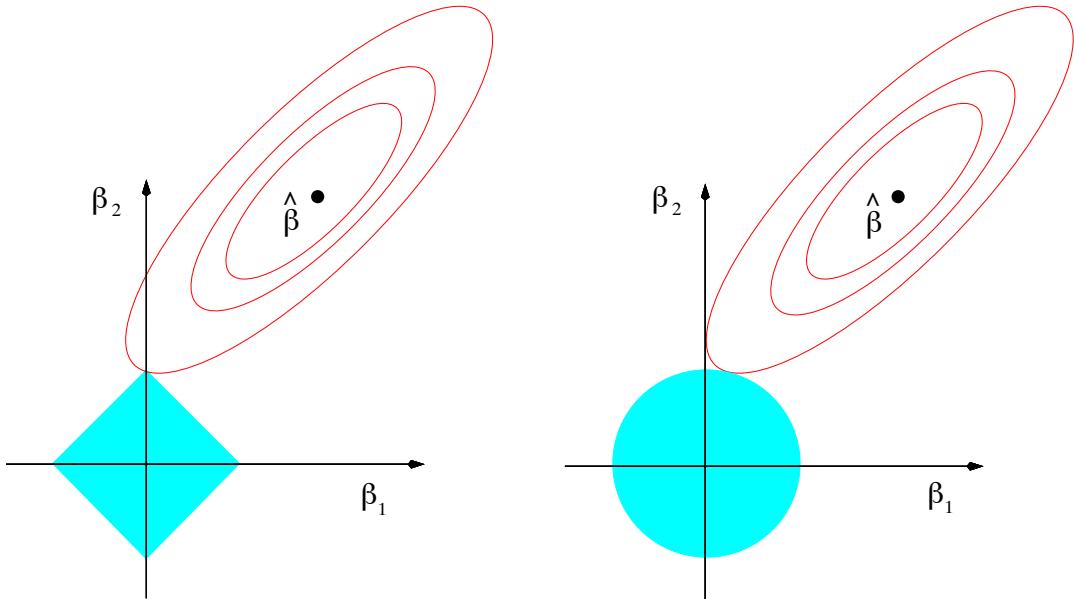


Figure 1: The “classic” illustration comparing lasso and ridge constraints. From Chapter 3 of [Hastie et al. \(2009\)](#)

Intuition can also be drawn from the orthogonal case. When X is orthogonal, it is not hard to show that the solutions of the penalized problems (4), (5), (6) are

$$\hat{\beta}^{\text{subset}} = H_{\sqrt{2\lambda}}(X^T y), \quad \hat{\beta}^{\text{lasso}} = S_\lambda(X^T y), \quad \hat{\beta}^{\text{ridge}} = \frac{X^T y}{1 + 2\lambda}$$

respectively, where $H_t(\cdot), S_t(\cdot)$ are the componentwise hard- and soft-thresholding functions at the level t . We see several revealing properties: subset selection and lasso solutions exhibit sparsity when the componentwise least squares coefficients (inner products $X^T y$) are small enough; the lasso solution exhibits shrinkage, in that large enough least squares coefficients are shrunken towards zero by λ ; the ridge regression solution is never sparse and compared to the lasso, preferentially shrinkage the larger least squares coefficients even more

2.4 Convexity

The lasso and ridge regression problems (2), (3) have another very important property: they are convex optimization problems. Best subset selection (1) is not, in fact it is very far from being convex. Consider using the norm $\|\beta\|_p$ as a penalty. Sparsity requires $p \leq 1$ and convexity requires $p \geq 1$. The only norm that gives sparsity and convexity is $p = 1$. The appendix has a brief review of convexity.

2.5 Theory For Subset Selection

Despite its computational intractability, best subset selection has some attractive risk properties. A classic result is due to [Foster & George \(1994\)](#), on the in-sample risk of best subset selection in penalized form (4), which we will paraphrase here. First, we raise a very simple point: if A denotes the support (also called the active set) of the subset selection solution $\hat{\beta}$ in (4)—meaning that $\hat{\beta}_j = 0$ for all $j \notin A$, and denoted $A = \text{supp}(\hat{\beta})$ —then we have

$$\begin{aligned}\hat{\beta}_A &= (X_A^T X_A)^{-1} X_A^T y, \\ \hat{\beta}_{-A} &= 0.\end{aligned}\tag{7}$$

Here and throughout we write X_A for the columns of matrix X in a set A , and x_A for the components of a vector x in A . We will also use X_{-A} and x_{-A} for the columns or components not in A . The observation in (7) follows from the fact that, given the support set A , the ℓ_0 penalty term in the subset selection criterion doesn't depend on the actual magnitudes of the coefficients (it contributes a constant factor), so the problem reduces to least squares.

Now, consider a standard linear model as with X fixed, and $\epsilon \sim N(0, \sigma^2 I)$. Suppose that the underlying coefficients have support $S = \text{supp}(\beta_0)$, and $s_0 = |S|$. Then, the estimator given by least squares on S , i.e.,

$$\begin{aligned}\hat{\beta}_S^{\text{oracle}} &= (X_S^T X_S)^{-1} X_S^T y, \\ \hat{\beta}_{-S}^{\text{oracle}} &= 0.\end{aligned}$$

is is called *oracle estimator*, and as we know from our previous calculations, has in-sample risk

$$\frac{1}{n} \|X \hat{\beta}^{\text{oracle}} - X \beta_0\|_2^2 = \sigma^2 \frac{s_0}{n}.$$

Foster & George (1994) consider this setup, and compare the risk of the best subset selection estimator $\hat{\beta}$ in (4) to the oracle risk of $\sigma^2 s_0/n$. They show that, if we choose $\lambda \asymp \sigma^2 \log p$, then the best subset selection estimator satisfies

$$\frac{\mathbb{E}\|X\hat{\beta} - X\beta_0\|_2^2/n}{\sigma^2 s_0/n} \leq 4 \log p + 2 + o(1), \quad (8)$$

as $n, p \rightarrow \infty$. This holds without any conditions on the predictor matrix X . Moreover, they prove the lower bound

$$\inf_{\hat{\beta}} \sup_{X, \beta_0} \frac{\mathbb{E}\|X\hat{\beta} - X\beta_0\|_2^2/n}{\sigma^2 s_0/n} \geq 2 \log p - o(\log p),$$

where the infimum is over all estimators $\hat{\beta}$, and the supremum is over all predictor matrices X and underlying coefficients with $\|\beta_0\|_0 = s_0$. Hence, in terms of rate, best subset selection achieves the optimal risk inflation over the oracle risk.

Returning to what was said above, the kicker is that we can't really compute the best subset selection estimator for even moderately-sized problems. As we will in the following, the lasso provides a similar risk inflation guarantee, though under considerably stronger assumptions.

Lastly, it is worth remarking that even if we *could* compute the subset selection estimator at scale, it's not at all clear that we would want to use this in place of the lasso. (Many people assume that we would.) We must remind ourselves that theory provides us an understanding of the performance of various estimators under typically idealized conditions, and it doesn't tell the complete story. It could be the case that the lack of shrinkage in the subset selection coefficients ends up being harmful in practical situations, in a signal-to-noise regime, and yet the lasso could still perform favorably in such settings.

Update. Some nice recent work in optimization (Bertsimas et al. 2016) shows that we can cast best subset selection as a mixed integer quadratic program, and proposes to solve it (in general this means approximately, though with a certified bound on the duality gap) with an industry-standard mixed integer optimization package like Gurobi. However, in a recent paper, Hastie, Tibshirani and Tibshirani (arXiv:1707.08692) show that best subset selection does not do well statistically unless there is an extremely high signal to noise ratio.

3 Basic properties and geometry of the lasso

3.1 Ridge regression and the elastic net

A quick refresher: the ridge regression problem (6) is always strictly convex (assuming $\lambda > 0$), due to the presence of the squared ℓ_2 penalty $\|\beta\|_2^2$. To be clear, this is true regardless of X , and so the ridge regression solution is always well-defined, and is in fact given in closed-form by $\hat{\beta} = (X^T X + 2\lambda I)^{-1} X^T y$.

3.2 Lasso

Now we turn to subgradient optimality (sometimes called the KKT conditions) for the lasso problem in (5). They tell us that any lasso solution $\hat{\beta}$ must satisfy

$$X^T(y - X\hat{\beta}) = \lambda s, \quad (9)$$

where $s \in \partial\|\hat{\beta}\|_1$, a subgradient of the ℓ_1 norm evaluated at $\hat{\beta}$. Precisely, this means that

$$s_j \in \begin{cases} \{+1\} & \hat{\beta}_j > 0 \\ \{-1\} & \hat{\beta}_j < 0 \\ [-1, 1] & \hat{\beta}_j = 0, \end{cases} \quad j = 1, \dots, p. \quad (10)$$

From (9) we can read off a straightforward but important fact: even though the solution $\hat{\beta}$ may not be uniquely determined, the optimal subgradient s is a function of the unique fitted value $X\hat{\beta}$ (assuming $\lambda > 0$), and hence is itself unique.

Now from (10), note that the uniqueness of s implies that any two lasso solutions must have the same signs on the overlap of their supports. That is, it cannot happen that we find two different lasso solutions $\hat{\beta}$ and $\tilde{\beta}$ with $\hat{\beta}_j > 0$ but $\tilde{\beta}_j < 0$ for some j , and hence we have no problem interpreting the signs of components of lasso solutions.

Let's assume henceforth that the columns of X are in general position (and we are looking at a nontrivial end of the path, with $\lambda > 0$), so the lasso solution $\hat{\beta}$ is unique. Let $A = \text{supp}(\hat{\beta})$ be the lasso active set, and let $s_A = \text{sign}(\hat{\beta}_A)$ be the signs of active coefficients. From the subgradient conditions (9), (10), we know that

$$X_A^T(y - X_A\hat{\beta}_A) = \lambda s_A,$$

and solving for $\hat{\beta}_A$ gives

$$\begin{aligned} \hat{\beta}_A &= (X_A^T X_A)^{-1} (X_A^T y - \lambda s_A), \\ \hat{\beta}_{-A} &= 0 \end{aligned} \quad (11)$$

(where recall we know that $X_A^T X_A$ is invertible because X has columns in general position). We see that the active coefficients $\hat{\beta}_A$ are given by taking the least squares coefficients on X_A , $(X_A^T X_A)^{-1} X_A^T y$, and shrinking them by an amount $\lambda(X_A^T X_A)^{-1} s_A$. Contrast this to, e.g., the subset selection solution in (7), where there is no such shrinkage.

Now, how about this so-called shrinkage term $(X_A^T X_A)^{-1} X_A^T y$? Does it always act by moving each one of the least squares coefficients $(X_A^T X_A)^{-1} X_A^T y$ towards zero? Indeed, this is not always the case, and one can find empirical examples where a lasso coefficient is actually larger (in magnitude) than the corresponding least squares coefficient on the active set. Of course, we also know that this is due to the correlations

between active variables, because when X is orthogonal, as we've already seen, this never happens.

On the other hand, it is always the case that the lasso solution has a strictly smaller ℓ_1 norm than the least squares solution on the active set, and in this sense, we are (perhaps) justified in always referring to $(X_A^T X_A)^{-1} X_A^T y$ as a shrinkage term. To see this, note that, for any vector b , $\|b\|_1 = s^T b$ where s is the vector of signs of b . So $\|\widehat{\beta}\|_1 = s^T \widehat{\beta} = s_A^T \widehat{\beta}_A$ and so

$$\|\widehat{\beta}\|_1 = s_A^T (X_A^T X_A)^{-1} X_A^T y - \lambda s_A^T (X_A^T X_A)^{-1} s_A < \|(X_A^T X_A)^{-1} X_A^T y\|_1. \quad (12)$$

The first term is less than or equal to $\|(X_A^T X_A)^{-1} X_A^T y\|_1$, and the term we are subtracting is strictly negative (because $(X_A^T X_A)^{-1}$ is positive definite).

4 Theoretical analysis of the lasso

4.1 Slow rates

There has been an enormous amount theoretical work analyzing the performance of the lasso. Some references (warning: a highly incomplete list) are [Greenshtein & Ritov \(2004\)](#), [Fuchs \(2005\)](#), [Donoho \(2006\)](#), [Candes & Tao \(2006\)](#), [Meinshausen & Bühlmann \(2006\)](#), [Zhao & Yu \(2006\)](#), [Candes & Plan \(2009\)](#), [Wainwright \(2009\)](#); a helpful text for these kind of results is [Bühlmann & van de Geer \(2011\)](#).

We begin by stating what are called *slow rates* for the lasso estimator. Most of the proofs are simple enough that they are given below. These results don't place any real assumptions on the predictor matrix X , but deliver slow(er) rates for the risk of the lasso estimator than what we would get under more assumptions, hence their name.

We will assume the standard linear model with X fixed, and $\epsilon \sim N(0, \sigma^2)$. We will also assume that $\|X_j\|_2^2 \leq n$, for $j = 1, \dots, p$. That the errors are Gaussian can be easily relaxed to sub-Gaussianity.

The lasso estimator in bound form (2) is particularly easy to analyze. Suppose that we choose $t = \|\beta_0\|_1$ as the tuning parameter. Then, simply by virtue of optimality of the solution $\widehat{\beta}$ in (2), we find that

$$\|y - X\widehat{\beta}\|_2^2 \leq \|y - X\beta_0\|_2^2,$$

or, expanding and rearranging,

$$\|X\widehat{\beta} - X\beta_0\|_2^2 \leq 2\langle \epsilon, X\widehat{\beta} - X\beta_0 \rangle.$$

Here we denote $\langle a, b \rangle = a^T b$. The above is sometimes called the *basic inequality* (for the lasso in bound form). Now, rearranging the inner product, using Holder's inequality, and recalling the choice of bound parameter:

$$\|X\widehat{\beta} - X\beta_0\|_2^2 \leq 2\langle X^T \epsilon, \widehat{\beta} - \beta_0 \rangle \leq 4\|\beta_0\|_1 \|X^T \epsilon\|_\infty.$$

Notice that $\|X^T \epsilon\|_\infty = \max_{j=1,\dots,p} |X_j^T \epsilon|$ is a maximum of p Gaussians, each with mean zero and variance upper bounded by $\sigma^2 n$. By a standard maximal inequality for Gaussians, for any $\delta > 0$,

$$\max_{j=1,\dots,p} |X_j^T \epsilon| \leq \sigma \sqrt{2n \log(ep/\delta)},$$

with probability at least $1 - \delta$. Plugging this to the second-to-last display and dividing by n , we get the finite-sample result for the lasso estimator

$$\frac{1}{n} \|X \hat{\beta} - X \beta_0\|_2^2 \leq 4\sigma \|\beta_0\|_1 \sqrt{\frac{2 \log(ep/\delta)}{n}}, \quad (13)$$

with probability at least $1 - \delta$.

The high-probability result (13) implies an in-sample risk bound of

$$\frac{1}{n} \mathbb{E} \|X \hat{\beta} - X \beta_0\|_2^2 \lesssim \|\beta_0\|_1 \sqrt{\frac{\log p}{n}}.$$

Compare to this with the risk bound (8) for best subset selection, which is on the (optimal) order of $s_0 \log p/n$ when β_0 has s_0 nonzero components. If each of the nonzero components here has constant magnitude, then above risk bound for the lasso estimator is on the order of $s_0 \sqrt{\log p/n}$, which is much slower.

Predictive risk. Instead of in-sample risk, we might also be interested in out-of-sample risk, as after all that reflects actual (out-of-sample) predictions. In least squares, recall, we saw that out-of-sample risk was generally higher than in-sample risk. The same is true for the lasso Chatterjee (2013) gives a nice, simple analysis of out-of-sample risk for the lasso. He assumes that $x_0, x_i, i = 1, \dots, n$ are i.i.d. from an arbitrary distribution supported on a compact set in \mathbb{R}^p , and shows that the lasso estimator in bound form (2) with $t = \|\beta_0\|_1$ has out-of-sample risk satisfying

$$\mathbb{E}(x_0^T \hat{\beta} - x_0^T \beta)^2 \lesssim \|\beta_0\|_1^2 \sqrt{\frac{\log p}{n}}.$$

The proof is not much more complicated than the above, for the in-sample risk, and reduces to a clever application of Hoeffding's inequality, though we omit it for brevity. Note here the dependence on $\|\beta_0\|_1^2$, rather than $\|\beta_0\|_1$ as in the in-sample risk. This agrees with the analysis we did in the previous set of notes where we did not assume the linear model. (Only the interpretation changes.)

Oracle inequality. If we don't want to assume linearity of the mean then we can still derive an *oracle inequality* that characterizes the risk of the lasso estimator in excess of the risk of the best linear predictor. For this part only, assume the more general model

$$y = \mu(X) + \epsilon,$$

with an arbitrary mean function $\mu(X)$, and normal errors $\epsilon \sim N(0, \sigma^2)$. We will analyze the bound form lasso estimator (2) for simplicity. By optimality of $\widehat{\beta}$, for any other $\widetilde{\beta}$ feasible for the lasso problem in (2), it holds that¹

$$\langle X^T(y - X\widehat{\beta}), \widetilde{\beta} - \widehat{\beta} \rangle \leq 0. \quad (14)$$

Rearranging gives

$$\langle \mu(X) - X\widehat{\beta}, X\widetilde{\beta} - X\widehat{\beta} \rangle \leq \langle X^T\epsilon, \widehat{\beta} - \widetilde{\beta} \rangle. \quad (15)$$

Now using the polarization identity $\|a\|_2^2 + \|b\|_2^2 - \|a - b\|_2^2 = 2\langle a, b \rangle$,

$$\|X\widehat{\beta} - \mu(X)\|_2^2 + \|X\widehat{\beta} - X\widetilde{\beta}\|_2^2 \leq \|X\widetilde{\beta} - \mu(X)\|_2^2 + 2\langle X^T\epsilon, \widehat{\beta} - \widetilde{\beta} \rangle,$$

and from the exact same arguments as before, it holds that

$$\frac{1}{n}\|X\widehat{\beta} - \mu(X)\|_2^2 + \frac{1}{n}\|X\widehat{\beta} - X\widetilde{\beta}\|_2^2 \leq \frac{1}{n}\|X\widetilde{\beta} - \mu(X)\|_2^2 + 4\sigma t \sqrt{\frac{2\log(ep/\delta)}{n}},$$

with probability at least $1 - \delta$. This holds simultaneously over all $\widetilde{\beta}$ with $\|\widetilde{\beta}\|_1 \leq t$. Thus, we may write, with probability $1 - \delta$,

$$\frac{1}{n}\|X\widehat{\beta} - \mu(X)\|_2^2 \leq \left\{ \inf_{\|\widetilde{\beta}\|_1 \leq t} \frac{1}{n}\|X\widetilde{\beta} - \mu(X)\|_2^2 \right\} + 4\sigma t \sqrt{\frac{2\log(ep/\delta)}{n}}.$$

Also if we write $X\widetilde{\beta}^{\text{best}}$ as the best linear predictor of ℓ_1 at most t , achieving the infimum on the right-hand side (which we know exists, as we are minimizing a continuous function over a compact set), then

$$\frac{1}{n}\|X\widehat{\beta} - X\widetilde{\beta}^{\text{best}}\|_2^2 \leq 4\sigma t \sqrt{\frac{2\log(ep/\delta)}{n}},$$

with probability at least $1 - \delta$

4.2 Fast rates

Under **very** strong assumptions we can get faster rates. For example, if we assume that X satisfies the *restricted eigenvalue condition* with constant $\phi_0 > 0$, i.e.,

$$\begin{aligned} \frac{1}{n}\|Xv\|_2^2 &\geq \phi_0^2\|v\|_2^2 \quad \text{for all subsets } J \subseteq \{1, \dots, p\} \text{ such that } |J| = s_0 \\ &\text{and all } v \in \mathbb{R}^p \text{ such that } \|v_{J^c}\|_1 \leq 3\|v_J\|_1 \end{aligned} \quad (16)$$

¹ To see this, consider minimizing a convex function $f(x)$ over a convex set C . Let \widehat{x} be a minimizer. Let $z \in C$ be any other point in C . If we move away from the solution \widehat{x} we can only increase $f(\widehat{x})$. In other words, $\langle \nabla f(\widehat{x}), z - \widehat{x} \rangle \geq 0$.

then

$$\|\hat{\beta} - \beta_0\|_2^2 \lesssim \frac{s_0 \log p}{n\phi_0^2} \quad (17)$$

with probability tending to 1. (This condition can be slightly weakened, but not much.) The condition is unlikely to hold in any real problem. Nor is it checkable. The proof is in the appendix.

4.3 Support recovery

Here we discuss results on support recovery of the lasso estimator. There are a few versions of support recovery results and again [Buhlmann & van de Geer \(2011\)](#) is a good place to look for a thorough coverage. Here we describe a result due to [Wainwright \(2009\)](#), who introduced a proof technique called the *primal-dual witness method*. The assumptions are even stronger (and less believable) than in the previous section. In addition to the previous assumptions we need:

Mutual incoherence: for some $\gamma > 0$, we have

$$\|(X_S^T X_S)^{-1} X_S^T X_j\|_1 \leq 1 - \gamma, \quad \text{for } j \notin S,$$

Minimum eigenvalue: for some $C > 0$, we have

$$\Lambda_{\min}\left(\frac{1}{n} X_S^T X_S\right) \geq C,$$

where $\Lambda_{\min}(A)$ denotes the minimum eigenvalue of a matrix A

Minimum signal:

$$\beta_{0,\min} = \min_{j \in S} |\beta_{0,j}| \geq \lambda \|(X_S^T X_S)^{-1}\|_\infty + \frac{4\gamma\lambda}{\sqrt{C}},$$

where $\|A\|_\infty = \max_{i=1, \dots, m} \sum_{j=1}^q |A_{ij}|$ denotes the ℓ_∞ norm of an $m \times q$ matrix A

Under these assumptions, one can show that, if λ is chosen just right, then

$$P(\text{support}(\hat{\beta}) = \text{support}(\beta)) \rightarrow 1. \quad (18)$$

The proof is in the appendix.

References

- Beale, E. M. L., Kendall, M. G. & Mann, D. W. (1967), ‘The discarding of variables in multivariate analysis’, *Biometrika* **54**(3/4), 357–366.
- Bertsimas, D., King, A. & Mazumder, R. (2016), ‘Best subset selection via a modern optimization lens’, *The Annals of Statistics* **44**(2), 813–852.

- Buhlmann, P. & van de Geer, S. (2011), *Statistics for High-Dimensional Data*, Springer.
- Candes, E. J. & Plan, Y. (2009), ‘Near ideal model selection by ℓ_1 minimization’, *Annals of Statistics* **37**(5), 2145–2177.
- Candes, E. J. & Tao, T. (2006), ‘Near optimal signal recovery from random projections: Universal encoding strategies?’, *IEEE Transactions on Information Theory* **52**(12), 5406–5425.
- Chatterjee, S. (2013), Assumptionless consistency of the lasso. arXiv: 1303.5817.
- Chen, S., Donoho, D. L. & Saunders, M. (1998), ‘Atomic decomposition for basis pursuit’, *SIAM Journal on Scientific Computing* **20**(1), 33–61.
- Donoho, D. L. (2006), ‘Compressed sensing’, *IEEE Transactions on Information Theory* **52**(12), 1289–1306.
- Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), ‘Least angle regression’, *Annals of Statistics* **32**(2), 407–499.
- Foster, D. & George, E. (1994), ‘The risk inflation criterion for multiple regression’, *The Annals of Statistics* **22**(4), 1947–1975.
- Fuchs, J. J. (2005), ‘Recovery of exact sparse representations in the presence of bounded noise’, *IEEE Transactions on Information Theory* **51**(10), 3601–3608.
- Greenshtein, E. & Ritov, Y. (2004), ‘Persistence in high-dimensional linear predictor selection and the virtue of overparametrization’, *Bernoulli* **10**(6), 971–988.
- Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer. Second edition.
- Hastie, T., Tibshirani, R. & Wainwright, M. (2015), *Statistical Learning with Sparsity: the Lasso and Generalizations*, Chapman & Hall.
- Hocking, R. R. & Leslie, R. N. (1967), ‘Selection of the best subset in regression analysis’, *Technometrics* **9**(4), 531–540.
- Hoerl, A. & Kennard, R. (1970), ‘Ridge regression: biased estimation for nonorthogonal problems’, *Technometrics* **12**(1), 55–67.
- Meinshausen, N. & Buhlmann, P. (2006), ‘High-dimensional graphs and variable selection with the lasso’, *The Annals of Statistics* **34**(3), 1436–1462.
- Osborne, M., Presnell, B. & Turlach, B. (2000a), ‘A new approach to variable selection in least squares problems’, *IMA Journal of Numerical Analysis* **20**(3), 389–404.

- Osborne, M., Presnell, B. & Turlach, B. (2000b), ‘On the lasso and its dual’, *Journal of Computational and Graphical Statistics* **9**(2), 319–337.
- Raskutti, G., Wainwright, M. J. & Yu, B. (2011), ‘Minimax rates of estimation for high-dimensional linear regression over ℓ_q -balls’, *IEEE Transactions on Information Theory* **57**(10), 6976–6994.
- Tibshirani, R. (1996), ‘Regression shrinkage and selection via the lasso’, *Journal of the Royal Statistical Society: Series B* **58**(1), 267–288.
- van de Geer, S. & Bühlmann, P. (2009), ‘On the conditions used to prove oracle results for the lasso’, *Electronic Journal of Statistics* **3**, 1360–1392.
- Wainwright, M. (2017), *High-Dimensional Statistics: A Non-Asymptotic View*, Cambridge University Press. To appear.
- Wainwright, M. J. (2009), ‘Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (lasso)’, *IEEE Transactions on Information Theory* **55**(5), 2183–2202.
- Zhao, P. & Yu, B. (2006), ‘On model selection consistency of lasso’, *Journal of Machine Learning Research* **7**, 2541–2564.

5 Appendix: Convexity

It is convexity that allows to equate (2), (5), and (3), (6) (and yes, the penalized forms are convex problems too). It is also convexity that allows us to both efficiently solve, and in some sense, precisely understand the nature of the lasso and ridge regression solutions

Here is a (far too quick) refresher/introduction to basic convex analysis and convex optimization. Recall that a set $C \subseteq \mathbb{R}^n$ is called *convex* if for any $x, y \in C$ and $t \in [0, 1]$, we have

$$tx + (1 - t)y \in C,$$

i.e., the line segment joining x, y lies entirely in C . A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is called *convex* if its domain $\text{dom}(f)$ is convex, and for any $x, y \in \text{dom}(f)$ and $t \in [0, 1]$,

$$f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y),$$

i.e., the function lies below the line segment joining its evaluations at x and y . A function is called *strictly convex* if this same inequality holds strictly for $x \neq y$ and $t \in (0, 1)$

E.g., lines, rays, line segments, linear spaces, affine spaces, hyperplanes, halfspaces, polyhedra, norm balls are all convex sets

E.g., affine functions $a^T x + b$ are convex and concave, quadratic functions $x^T Q x + b^T x + c$ are convex if $Q \succeq 0$ and strictly convex if $Q \succ 0$, norms are convex

Formally, an *optimization problem* is of the form

$$\begin{aligned} & \min_{x \in D} f(x) \\ \text{subject to } & h_i(x) \leq 0, \quad i = 1, \dots, m \\ & \ell_j(x) = 0, \quad j = 1, \dots, r \end{aligned}$$

Here $D = \text{dom}(f) \cap \bigcap_{i=1}^m \text{dom}(h_i) \cap \bigcap_{j=1}^r \text{dom}(\ell_j)$ is the common domain of all functions. A *convex optimization problem* is an optimization problem in which all functions f, h_1, \dots, h_m are convex, and all functions ℓ_1, \dots, ℓ_r are affine. (Think: why affine?) Hence, we can express it as

$$\begin{aligned} & \min_{x \in D} f(x) \\ \text{subject to } & h_i(x) \leq 0, \quad i = 1, \dots, m \\ & Ax = b \end{aligned}$$

Why is a convex optimization problem so special? The short answer: because *any local minimizer is a global minimizer*. To see this, suppose that x is feasible for the convex problem formulation above and there exists some $R > 0$ such that

$$f(x) \leq f(y) \quad \text{for all feasible } y \text{ with } \|x - y\|_2 \leq R.$$

Such a point x is called a local minimizer. For the sake of contradiction, suppose that x was not a global minimizer, i.e., there exists some feasible z such that $f(z) < f(x)$. By convexity of the constraints (and the domain D), the point $tz + (1-t)x$ is feasible for any $0 \leq t \leq 1$. Furthermore, by convexity of f ,

$$f(tz + (1-t)x) \leq tf(z) + (1-t)f(x) < f(x)$$

for any $0 < t < 1$. Lastly, we can choose $t > 0$ small enough so that $\|x - (tz + (1-t)x)\|_2 = t\|x - z\|_2 \leq R$, and we obtain a contradiction

Algorithmically, this is a very useful property, because it means if we keep “going downhill”, i.e., reducing the achieved criterion value, and we stop when we can’t do so anymore, then we’ve hit the global solution

Convex optimization problems are also special because they come with a beautiful theory of beautiful convex duality and optimality, which gives us a way of understanding the solutions. We won’t have time to cover any of this, but we’ll mention what subgradient optimality looks like for the lasso

Just based on the definitions, it is not hard to see that (2), (3), (5), (6) are convex problems, but (1), (4) are not. In fact, the latter two problems are known to be NP-hard, so they are in a sense even the worst kind of nonconvex problem

6 Appendix: Geometry of the solutions

One undesirable feature of the best subset selection solution (7) is the fact that it behaves discontinuously with y . As we change y , the active set A must change at some point, and the coefficients will jump discontinuously, because we are just doing least squares onto the active set. So, does the same thing happen with the lasso solution (11)? The answer is not immediately clear. Again, as we change y , the active set A must change at some point; but if the shrinkage term were defined “just right”, then perhaps the coefficients of variables to leave the active set would gracefully and continuously drop to zero, and coefficients of variables to enter the active set would continuously move from zero. This would make whole the lasso solution continuous. Fortunately, this is indeed the case, and the lasso solution $\hat{\beta}$ is continuous as a function of y . It might seem a daunting task to prove this, but a certain perspective using convex geometry provides a very simple proof. The geometric perspective in fact proves that the lasso fit $X\hat{\beta}$ is nonexpansive in y , i.e., 1-Lipschitz continuous, which is a very strong form of continuity. Define the convex polyhedron $C = \{u : \|X^T u\|_\infty \leq \lambda\} \subseteq \mathbb{R}^n$. Some simple manipulations of the KKT conditions show that the lasso fit is given by

$$X\hat{\beta} = (I - P_C)(y),$$

the residual from projecting y onto C . A picture to show this (just look at the left panel for now) is given in Figure 2.

The projection onto any convex set is nonexpansive, i.e., $\|P_C(y) - P_C(y')\|_2 \leq \|y - y'\|_2$ for any y, y' . This should be visually clear from the picture. Actually, the same is true with the residual map: $I - P_C$ is also nonexpansive, and hence the lasso fit is 1-Lipschitz continuous. Viewing the lasso fit as the residual from projection onto a convex polyhedron is actually an even more fruitful perspective. Write this polyhedron as

$$C = (X^T)^{-1}\{v : \|v\|_\infty \leq \lambda\},$$

where $(X^T)^{-1}$ denotes the preimage operator under the linear map X^T . The set $\{v : \|v\|_\infty \leq \lambda\}$ is a hypercube in \mathbb{R}^p . Every face of this cube corresponds to a subset $A \subseteq \{1, \dots, p\}$ of dimensions (that achieve the maximum value $|\lambda|$) and signs $s_A \in \{-1, 1\}^{|A|}$ (that tell which side of the cube the face will lie on, for each dimension). Now, the faces of C are just faces of $\{v : \|v\|_\infty \leq \lambda\}$ run through the (linear) preimage transformation, so each face of C can also indexed by a set $A \subseteq \{1, \dots, p\}$ and signs $s_A \in \{-1, 1\}^{|A|}$. The picture in Figure 2 attempts to convey this relationship with the colored black face in each of the panels.

Now imagine projecting y onto C ; it will land on some face. We have just argued that this face corresponds to a set A and signs s_A . One can show that this set A is exactly the active set of the lasso solution at y , and s_A are exactly the active signs. The size of the active set $|A|$ is the co-dimension of the face. Looking at the picture: we can see that as we wiggle y around, it will project to the same face. From the

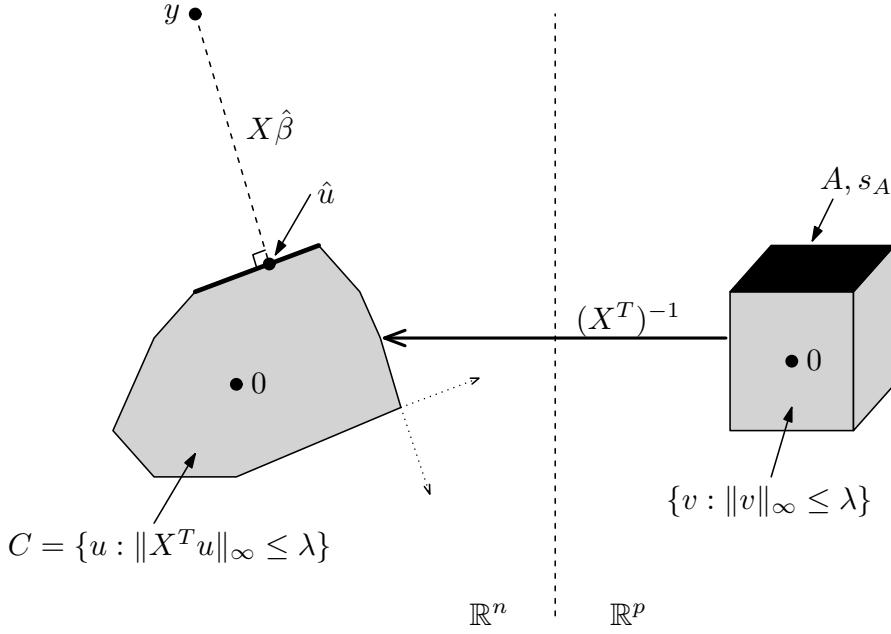


Figure 2: A geometric picture of the lasso solution. The left panel shows the polyhedron underlying all lasso fits, where each face corresponds to a particular combination of active set A and signs s ; the right panel displays the “inverse” polyhedron, where the dual solutions live

correspondence between faces and active set and signs of lasso solutions, this means that A, s_A do not change as we perturb y , i.e., they are locally constant. But this isn’t true for all points y , e.g., if y lies on one of the rays emanating from the lower right corner of the polyhedron in the picture, then we can see that small perturbations of y do actually change the face that it projects to, which invariably changes the active set and signs of the lasso solution. However, this is somewhat of an exceptional case, in that such points can be form a of Lebesgue measure zero, and therefore we can assure ourselves that the active set and signs A, s_A are locally constant for almost every y .

From the lasso KKT conditions (9), (10), it is possible to compute the lasso solution in (5) as a function of λ , which we will write as $\hat{\beta}(\lambda)$, for all values of the tuning parameter $\lambda \in [0, \infty]$. This is called the *regularization path* or *solution path* of the problem (5). Path algorithms like the one we will describe below are not always possible; the reason that this ends up being feasible for the lasso problem (5) is that the solution path $\hat{\beta}(\lambda)$, $\lambda \in [0, \infty]$ turns out to be a piecewise linear, continuous function of λ . Hence, we only need to compute and store the *knots* in this path, which we will denote by $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r \geq 0$, and the lasso solution at these knots. From this information, we can then compute the lasso solution at any value

of λ by linear interpolation.

The knots $\lambda_1 \geq \dots \geq \lambda_r$ in the solution path correspond to λ values at which the active set $A(\lambda) = \text{supp}(\hat{\beta}(\lambda))$ changes. As we decrease λ from ∞ to 0, the knots typically correspond to the point at which a variable enters the active set; this connects the lasso to an incremental variable selection procedure like forward stepwise regression. Interestingly though, as we decrease λ , a knot in the lasso path can also correspond to the point at which a variable leaves the active set. See Figure 3.

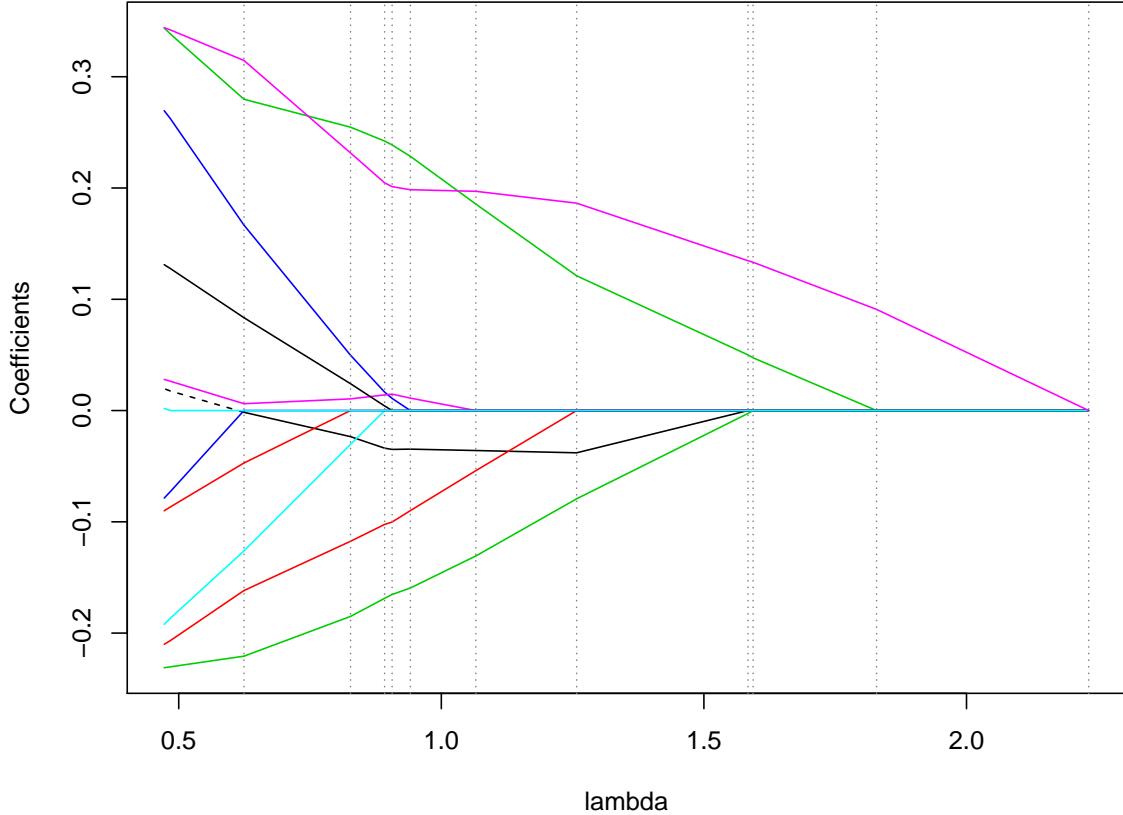


Figure 3: An example of the lasso path. Each colored line denotes a component of the lasso solution $\hat{\beta}_j(\lambda)$, $j = 1, \dots, p$ as a function of λ . The gray dotted vertical lines mark the knots $\lambda_1 \geq \lambda_2 \geq \dots$

The lasso solution path was described by Osborne et al. (2000a,b), Efron et al. (2004). Like the construction of all other solution paths that followed these seminal works, the lasso path is essentially given by an iterative or inductive verification of the KKT conditions; if we can maintain that the KKT conditions holds as we decrease λ , then we know we have a solution. The trick is to start at a value of λ at which the solution is trivial; for the lasso, this is $\lambda = \infty$, at which case we know the solution must be $\hat{\beta}(\infty) = 0$.

Why would the path be piecewise linear? The construction of the path from the

KKT conditions is actually rather technical (not difficult conceptually, but somewhat tedious), and doesn't shed insight onto this matter. But we can actually see it clearly from the projection picture in Figure 2.

As λ decreases from ∞ to 0, we are shrinking (by a multiplicative factor λ) the polyhedron onto which y is projected; let's write $C_\lambda = \{u : \|X^T u\|_\infty \leq \lambda\} = \lambda C_1$ to make this clear. Now suppose that y projects onto the relative interior of a certain face F of C_λ , corresponding to an active set A and signs s_A . As λ decreases, the point on the boundary of C_λ onto which y projects, call it $\hat{u}(\lambda) = P_{C_\lambda}(y)$, will move along the face F , and change linearly in λ (because we are equivalently just tracking the projection of y onto an affine space that is being scaled by λ). Thus, the lasso fit $X\hat{\beta}(\lambda) = y - \hat{u}(\lambda)$ will also behave linearly in λ . Eventually, as we continue to decrease λ , the projected point $\hat{u}(\lambda)$ will move to the relative boundary of the face F ; then, decreasing λ further, it will lie on a different, neighboring face F' . This face will correspond to an active set A' and signs $s_{A'}$ that (each) differ by only one element to A and s_A , respectively. It will then move linearly across F' , and so on.

Now we will walk through the technical derivation of the lasso path, starting at $\lambda = \infty$ and $\hat{\beta}(\infty) = 0$, as indicated above. Consider decreasing λ from ∞ , and continuing to set $\hat{\beta}(\lambda) = 0$ as the lasso solution. The KKT conditions (9) read

$$X^T y = \lambda s,$$

where s is a subgradient of the ℓ_1 norm evaluated at 0, i.e., $s_j \in [-1, 1]$ for every $j = 1, \dots, p$. For large enough values of λ , this is satisfied, as we can choose $s = X^T y / \lambda$. But this ceases to be a valid subgradient if we decrease λ past the point at which $\lambda = |X_j^T y|$ for some variable $j = 1, \dots, p$. In short, $\hat{\beta}(\lambda) = 0$ is the lasso solution for all $\lambda \geq \lambda_1$, where

$$\lambda_1 = \max_{j=1, \dots, p} |X_j^T y|. \quad (19)$$

What happens next? As we decrease λ from λ_1 , we know that we're going to have to change $\hat{\beta}(\lambda)$ from 0 so that the KKT conditions remain satisfied. Let j_1 denote the variable that achieves the maximum in (19). Since the subgradient was $|s_{j_1}| = 1$ at $\lambda = \lambda_1$, we see that we are "allowed" to make $\hat{\beta}_{j_1}(\lambda)$ nonzero. Consider setting

$$\begin{aligned} \hat{\beta}_{j_1}(\lambda) &= (X_{j_1}^T X_{j_1})^{-1} (X_{j_1}^T y - \lambda s_{j_1}) \\ \hat{\beta}_j(\lambda) &= 0, \quad \text{for all } j \neq j_1, \end{aligned} \quad (20)$$

as λ decreases from λ_1 , where $s_{j_1} = \text{sign}(X_{j_1}^T y)$. Note that this makes $\hat{\beta}(\lambda)$ a piecewise linear and continuous function of λ , so far. The KKT conditions are then

$$X_{j_1}^T \left(y - X_{j_1} (X_{j_1}^T X_{j_1})^{-1} (X_{j_1}^T y - \lambda s_{j_1}) \right) = \lambda s_{j_1},$$

which can be checked with simple algebra, and

$$\left| X_j^T \left(y - X_{j_1} (X_{j_1}^T X_{j_1})^{-1} (X_{j_1}^T y - \lambda s_{j_1}) \right) \right| \leq \lambda,$$

for all $j \neq j_1$. Recall that the above held with strict inequality at $\lambda = \lambda_1$ for all $j \neq j_1$, and by continuity of the constructed solution $\widehat{\beta}(\lambda)$, it should continue to hold as we decrease λ for at least a little while. In fact, it will hold until one of the piecewise linear paths

$$X_j^T(y - X_{j_1}(X_{j_1}^T X_{j_1})^{-1}(X_{j_1}^T y - \lambda s_{j_1})), \quad j \neq j_1$$

becomes equal to $\pm\lambda$, at which point we have to modify the solution because otherwise the implicit subgradient

$$s_j = \frac{X_j^T(y - X_{j_1}(X_{j_1}^T X_{j_1})^{-1}(X_{j_1}^T y - \lambda s_{j_1}))}{\lambda}$$

will cease to be in $[-1, 1]$. It helps to draw yourself a picture of this.

Thanks to linearity, we can compute the critical ‘‘hitting time’’ explicitly; a short calculation shows that, the lasso solution continues to be given by (20) for all $\lambda_1 \geq \lambda \geq \lambda_2$, where

$$\lambda_2 = \max_{j \neq j_1, s_j \in \{-1, 1\}}^+ \frac{X_j^T(I - X_{j_1}(X_{j_1}^T X_{j_1})^{-1} X_{j_1})y}{s_j - X_j^T X_{j_1}(X_{j_1}^T X_{j_1})^{-1} s_{j_1}}, \quad (21)$$

and \max^+ denotes the maximum over all of its arguments that are $< \lambda_1$.

To keep going: let j_2, s_2 achieve the maximum in (21). Let $A = \{j_1, j_2\}$, $s_A = (s_{j_1}, s_{j_2})$, and consider setting

$$\begin{aligned} \widehat{\beta}_A(\lambda) &= (X_A^T X_A)^{-1}(X_A^T y - \lambda s_A) \\ \widehat{\beta}_{-A}(\lambda) &= 0, \end{aligned} \quad (22)$$

as λ decreases from λ_2 . Again, we can verify the KKT conditions for a stretch of decreasing λ , but will have to stop when one of

$$X_j^T(y - X_A(X_A^T X_A)^{-1}(X_A^T y - \lambda s_A)), \quad j \notin A$$

becomes equal to $\pm\lambda$. By linearity, we can compute this next ‘‘hitting time’’ explicitly, just as before. Furthermore, though, we will have to check whether the active components of the computed solution in (22) are going to cross through zero, because past such a point, s_A will no longer be a proper subgradient over the active components. We can again compute this next ‘‘crossing time’’ explicitly, due to linearity. Therefore, we maintain that (22) is the lasso solution for all $\lambda_2 \geq \lambda \geq \lambda_3$, where λ_3 is the maximum of the next hitting time and the next crossing time. For convenience, the lasso path algorithm is summarized below.

As we decrease λ from a knot λ_k , we can rewrite the lasso coefficient update in Step 1 as

$$\begin{aligned} \widehat{\beta}_A(\lambda) &= \widehat{\beta}_A(\lambda_k) + (\lambda_k - \lambda)(X_A^T X_A)^{-1}s_A, \\ \widehat{\beta}_{-A}(\lambda) &= 0. \end{aligned} \quad (23)$$

We can see that we are moving the active coefficients in the direction $(\lambda_k - \lambda)(X_A^T X_A)^{-1} s_A$ for decreasing λ . In other words, the lasso fitted values proceed as

$$X\widehat{\beta}(\lambda) = X\widehat{\beta}(\lambda_k) + (\lambda_k - \lambda)X_A(X_A^T X_A)^{-1} s_A,$$

for decreasing λ . [Efron et al. \(2004\)](#) call $X_A(X_A^T X_A)^{-1} s_A$ the *equiangular direction*, because this direction, in \mathbb{R}^n , takes an equal angle with all $X_j \in \mathbb{R}^n$, $j \in A$.

For this reason, the lasso path algorithm in Algorithm ?? is also often referred to as the *least angle regression* path algorithm in “lasso mode”, though we have not mentioned this yet to avoid confusion. Least angle regression is considered as another algorithm by itself, where we skip Step 3 altogether. In words, Step 3 disallows any component path to cross through zero. The left side of the plot in Figure 3 visualizes the distinction between least angle regression and lasso estimates: the dotted black line displays the least angle regression component path, crossing through zero, while the lasso component path remains at zero.

Lastly, an alternative expression for the coefficient update in (23) (the update in Step 1) is

$$\begin{aligned}\widehat{\beta}_A(\lambda) &= \widehat{\beta}_A(\lambda_k) + \frac{\lambda_k - \lambda}{\lambda_k} (X_A^T X_A)^{-1} X_A^T r(\lambda_k), \\ \widehat{\beta}_{-A}(\lambda) &= 0,\end{aligned}\tag{24}$$

where $r(\lambda_k) = y - X_A \widehat{\beta}_A(\lambda_k)$ is the residual (from the fitted lasso model) at λ_k . This follows because, recall, $\lambda_k s_A$ are simply the inner products of the active variables with the residual at λ_k , i.e., $\lambda_k s_A = X_A^T (y - X_A \widehat{\beta}_A(\lambda_k))$. In words, we can see that the update for the active lasso coefficients in (24) is in the direction of the least squares coefficients of the residual $r(\lambda_k)$ on the active variables X_A .

7 Appendix: Fast Rates

Here is a proof of (17). There are many flavors of fast rates, and the conditions required are all very closely related. [van de Geer & Bühlmann \(2009\)](#) provides a nice review and discussion. Here we just discuss two such results, for simplicity.

Compatibility result. Assume that X satisfies the *compatibility condition* with respect to the true support set S , i.e., for some compatibility constant $\phi_0 > 0$,

$$\frac{1}{n} \|Xv\|_2^2 \geq \frac{\phi_0^2}{s_0} \|v_S\|_1^2 \quad \text{for all } v \in \mathbb{R}^p \text{ such that } \|v_{-S}\|_1 \leq 3\|v_S\|_1.\tag{25}$$

While this may look like an odd condition, we will see it being useful in the proof below, and we will also have some help interpreting it when we discuss the restricted eigenvalue condition shortly. Roughly, it means the (truly active) predictors can't be too correlated

Recall from our previous analysis for the lasso estimator in penalized form (5), we showed on an event E_δ of probability at least $1 - \delta$,

$$\|X\hat{\beta} - X\beta_0\|_2^2 \leq 2\sigma\sqrt{2n\log(ep/\delta)}\|\hat{\beta} - \beta_0\|_1 + 2\lambda(\|\beta_0\|_1 - \|\hat{\beta}\|_1).$$

Choosing λ large enough and applying the triangle inequality then gave us the slow rate we derived before. Now we choose λ just slightly larger (by a factor of 2): $\lambda \geq 2\sigma\sqrt{2n\log(ep/\delta)}$. The remainder of the analysis will be performed on the event E_δ and we will no longer make this explicit until the very end. Then

$$\begin{aligned} \|X\hat{\beta} - X\beta_0\|_2^2 &\leq \lambda\|\hat{\beta} - \beta_0\|_1 + 2\lambda(\|\beta_0\|_1 - \|\hat{\beta}\|_1) \\ &\leq \lambda\|\hat{\beta}_S - \beta_{0,S}\|_1 + \lambda\|\hat{\beta}_{-S}\|_1 + 2\lambda(\|\beta_0\|_1 - \|\hat{\beta}\|_1) \\ &\leq \lambda\|\hat{\beta}_S - \beta_{0,S}\|_1 + \lambda\|\hat{\beta}_{-S}\|_1 + 2\lambda(\|\beta_{0,S} - \hat{\beta}_S\|_1 - \|\hat{\beta}_{-S}\|_1) \\ &= 3\lambda\|\hat{\beta}_S - \beta_{0,S}\|_1 - \lambda\|\hat{\beta}_{-S}\|_1, \end{aligned}$$

where the two inequalities both followed from the triangle inequality, one application for each of the two terms, and we have used that $\hat{\beta}_{0,-S} = 0$. As $\|X\hat{\beta} - X\beta_0\|_2^2 \geq 0$, we have shown

$$\|\hat{\beta}_{-S} - \hat{\beta}_{0,-S}\|_1 \leq 3\|\hat{\beta}_S - \beta_{0,S}\|_1,$$

and thus we may apply the compatibility condition (25) to the vector $v = \hat{\beta} - \beta_0$. This gives us two bounds: one on the fitted values, and the other on the coefficients. Both start with the key inequality (from the second-to-last display)

$$\|X\hat{\beta} - X\beta_0\|_2^2 \leq 3\lambda\|\hat{\beta}_S - \beta_{0,S}\|_1. \quad (26)$$

For the fitted values, we upper bound the right-hand side of the key inequality (26),

$$\|X\hat{\beta} - X\beta_0\|_2^2 \leq 3\lambda\sqrt{\frac{s_0}{n\phi_0^2}}\|X\hat{\beta} - X\beta_0\|_2,$$

or dividing through both sides by $\|X\hat{\beta} - X\beta_0\|_2$, then squaring both sides, and dividing by n ,

$$\frac{1}{n}\|X\hat{\beta} - X\beta_0\|_2^2 \leq \frac{9s_0\lambda^2}{n^2\phi_0^2}.$$

Plugging in $\lambda = 2\sigma\sqrt{2n\log(ep/\delta)}$, we have shown that

$$\frac{1}{n}\|X\hat{\beta} - X\beta_0\|_2^2 \leq \frac{72\sigma^2 s_0 \log(ep/\delta)}{n\phi_0^2}, \quad (27)$$

with probability at least $1 - \delta$. Notice the similarity between (27) and (8): both provide us in-sample risk bounds on the order of $s_0 \log p/n$, but the bound for the lasso requires a strong compatibility assumption on the predictor matrix X , which roughly means the predictors can't be too correlated

For the coefficients, we lower bound the left-hand side of the key inequality (26),

$$\frac{n\phi_0^2}{s_0} \|\widehat{\beta}_S - \beta_{0,S}\|_1^2 \leq 3\lambda \|\widehat{\beta}_S - \beta_{0,S}\|_1,$$

so dividing through both sides by $\|\widehat{\beta}_S - \beta_{0,S}\|_1$, and recalling $\|\widehat{\beta}_{-S}\|_1 \leq 3\|\widehat{\beta}_S - \beta_{0,S}\|_1$, which implies by the triangle inequality that $\|\widehat{\beta} - \beta_0\|_1 \leq 4\|\widehat{\beta}_S - \beta_{0,S}\|_1$,

$$\|\widehat{\beta} - \beta_0\|_1 \leq \frac{12s_0\lambda}{n\phi_0^2}.$$

Plugging in $\lambda = 2\sigma\sqrt{2n\log(ep/\delta)}$, we have shown that

$$\|\widehat{\beta} - \beta_0\|_1 \leq \frac{24\sigma s_0}{\phi_0^2} \sqrt{\frac{2\log(ep/\delta)}{n}}, \quad (28)$$

with probability at least $1 - \delta$. This is a error bound on the order of $s_0\sqrt{\log p/n}$ for the lasso coefficients (in ℓ_1 norm)

Restricted eigenvalue result. Instead of compatibility, we may assume that X satisfies the *restricted eigenvalue condition* with constant $\phi_0 > 0$, i.e.,

$$\begin{aligned} \frac{1}{n} \|Xv\|_2^2 &\geq \phi_0^2 \|v\|_2^2 \quad \text{for all subsets } J \subseteq \{1, \dots, p\} \text{ such that } |J| = s_0 \\ &\quad \text{and all } v \in \mathbb{R}^p \text{ such that } \|v_{J^c}\|_1 \leq 3\|v_J\|_1. \end{aligned} \quad (29)$$

This produces essentially the same results as in (27), (28), but additionally, in the ℓ_2 norm,

$$\|\widehat{\beta} - \beta_0\|_2^2 \lesssim \frac{s_0 \log p}{n\phi_0^2}$$

with probability tending to 1

Note the similarity between (29) and the compatibility condition (25). The former is actually stronger, i.e., it implies the latter, because $\|\beta\|_2^2 \geq \|\beta_J\|_2^2 \geq \|\beta_J\|_1^2/s_0$. We may interpret the restricted eigenvalue condition roughly as follows: the requirement $(1/n)\|Xv\|_2^2 \geq \phi_0^2 \|v\|_2^2$ for all $v \in \mathbb{R}^n$ would be a lower bound of ϕ_0^2 on the smallest eigenvalue of $(1/n)X^T X$; we don't require this (as this would of course mean that X was full column rank, and couldn't happen when $p > n$), but instead that require that the same inequality hold for v that are "mostly" supported on small subsets J of variables, with $|J| = s_0$

8 Appendix: Support Recovery

Again we assume a standard linear model (??), with X fixed, subject to the scaling $\|X_j\|_2^2 \leq n$, for $j = 1, \dots, p$, and $\epsilon \sim N(0, \sigma^2)$. Denote by $S = \text{supp}(\beta_0)$ the true support set, and $s_0 = |S|$. Assume that X_S has full column rank

We aim to show that, at some value of λ , the lasso solution $\widehat{\beta}$ in (5) has an active set that exactly equals the true support set,

$$A = \text{supp}(\widehat{\beta}) = S,$$

with high probability. We actually aim to show that the signs also match,

$$\text{sign}(\widehat{\beta}_S) = \text{sign}(\beta_{0,S}),$$

with high probability. The primal-dual witness method basically plugs in the true support S into the KKT conditions for the lasso (9), (10), and checks when they can be verified

We start by breaking up (9) into two blocks, over S and S^c . Suppose that $\text{supp}(\widehat{\beta}) = S$ at a solution $\widehat{\beta}$. Then the KKT conditions become

$$X_S^T(y - X_S\widehat{\beta}_S) = \lambda s_S \quad (30)$$

$$X_{-S}^T(y - X_S\widehat{\beta}_S) = \lambda s_{-S}. \quad (31)$$

Hence, if we can satisfy the two conditions (30), (31) with a proper subgradient s , such that

$$s_S = \text{sign}(\beta_{0,S}) \quad \text{and} \quad \|s_{-S}\|_\infty = \max_{j \notin S} |s_j| < 1,$$

then we have met our goal: we have recovered a (unique) lasso solution whose active set is S , and whose active signs are $\text{sign}(\beta_{0,S})$

So, let's solve for $\widehat{\beta}_S$ in the first block (30). Just as we did in the work on basic properties of the lasso estimator, this yields

$$\widehat{\beta}_S = (X_S^T X_S)^{-1} (X_S^T y - \lambda \text{sign}(\beta_{0,S})), \quad (32)$$

where we have substituted $s_S = \text{sign}(\beta_{0,S})$. From (31), this implies that s_{-S} must satisfy

$$s_{-S} = \frac{1}{\lambda} X_{-S}^T (I - X_S (X_S^T X_S)^{-1} X_S^T) y + X_{-S}^T X_S (X_S^T X_S)^{-1} \text{sign}(\beta_{0,S}). \quad (33)$$

To lay it out, for concreteness, the primal-dual witness method proceeds as follows:

1. Solve for the lasso solution over the S components, $\widehat{\beta}_S$, as in (32), and set $\widehat{\beta}_{-S} = 0$
2. Solve for the subgradient over the S^c components, s_{-S} , as in (33)
3. Check that $\text{sign}(\widehat{\beta}_S) = \text{sign}(\beta_{0,S})$, and that $\|s_{-S}\|_\infty < 1$. If these two checks pass, then we have certified there is a (unique) lasso solution that exactly recovers the true support and signs

The success of the primal-dual witness method hinges on Step 3. We can plug in $y = X\beta_0 + \epsilon$, and rewrite the required conditions, $\text{sign}(\widehat{\beta}_S) = \text{sign}(\beta_{0,S})$ and $\|s_{-S}\|_\infty < 1$, as

$$\begin{aligned} \text{sign}(\beta_{0,j} + \Delta_j) &= \text{sign}(\beta_{0,j}), \text{ where} \\ \Delta_j &= e_j^T (X_S^T X_S)^{-1} (X_S^T \epsilon - \lambda \text{sign}(\beta_{0,S})), \text{ for all } j \in S, \end{aligned} \quad (34)$$

and

$$\left\| \frac{1}{\lambda} X_{-S}^T (I - X_S (X_S^T X_S)^{-1} X_S^T) \epsilon + X_{-S}^T X_S (X_S^T X_S)^{-1} \text{sign}(\beta_{0,S}) \right\|_\infty < 1. \quad (35)$$

As $\epsilon \sim N(0, \sigma^2 I)$, we see that the two required conditions have been reduced to statements about Gaussian random variables. The arguments we need to check these conditions actually are quite simple, but we will need to make assumptions on X and β_0 . These are:

With these assumptions in place on X and β_0 , let's first consider verifying (34), and examine Δ_S , whose components Δ_j , $j \in S$ are as defined in (34). We have

$$\|\Delta_S\|_\infty \leq \|(X_S^T X_S)^{-1} X_S^T \epsilon\|_\infty + \lambda \|(X_S^T X_S)^{-1}\|_\infty.$$

Note that $w = (X_S^T X_S)^{-1} X_S^T \epsilon$ is Gaussian with mean zero and covariance $\sigma^2 (X_S^T X_S)^{-1}$, so the variances of components of w are bounded by

$$\sigma^2 \Lambda_{\max}((X_S^T X_S)^{-1}) \leq \frac{\sigma^2 n}{C},$$

where we have used the minimum eigenvalue assumption. By a standard result on the maximum of Gaussians, for any $\delta > 0$, it holds with probability at least $1 - \delta$ that

$$\begin{aligned} \|\Delta_S\|_\infty &\leq \frac{\sigma}{\sqrt{C}} \sqrt{2n \log(es_0/\delta)} + \lambda \|(X_S^T X_S)^{-1}\|_\infty \\ &\leq \beta_{0,\min} + \underbrace{\frac{\gamma}{\sqrt{C}} \left(\frac{\sigma}{\gamma} \sqrt{2n \log(es_0/\delta)} - 4\lambda \right)}_a. \end{aligned}$$

where in the second line we used the minimum signal condition. As long as $a < 0$, we can see that the sign condition (34) is verified

Now, let's consider verifying (35). Using the mutual incoherence condition, we have

$$\left\| \frac{1}{\lambda} X_{-S}^T (I - X_S (X_S^T X_S)^{-1} X_S^T) \epsilon + X_{-S}^T X_S (X_S^T X_S)^{-1} \text{sign}(\beta_{0,S}) \right\|_\infty \leq \|z\|_\infty + (1 - \gamma),$$

where $z = (1/\lambda) X_{-S}^T (I - X_S (X_S^T X_S)^{-1} X_S^T) \epsilon = (1/\lambda) X_{-S}^T P_{X_S} \epsilon$, with P_{X_S} the projection matrix onto the column space of X_S . Notice that z is Gaussian with mean zero

and covariance $(\sigma^2/\lambda^2)X_{-S}^T P_{X_S} X_{-S}$, so the components of z have variances bounded by

$$\frac{\sigma^2 n}{\lambda^2} \Lambda_{\max}(P_{X_S}) \leq \frac{\sigma^2 n}{\lambda^2}.$$

Therefore, again by the maximal Gaussian inequality, for any $\delta > 0$, it holds with probability at least $1 - \delta$ that

$$\begin{aligned} \left\| \frac{1}{\lambda} X_{-S}^T (I - X_S(X_S^T X_S)^{-1} X_S^T) \epsilon + X_{-S}^T X_S (X_S^T X_S)^{-1} \text{sign}(\beta_{0,S}) \right\|_\infty \\ \leq \frac{\sigma}{\lambda} \sqrt{2n \log(e(p - s_0)/\delta)} + (1 - \gamma) \\ = 1 + \underbrace{\left(\frac{\sigma}{\lambda} \sqrt{2n \log(e(p - s_0)/\delta)} - \gamma \right)}_b, \end{aligned}$$

Thus as long as $b < 0$, we can see that the subgradient condition (35) is verified

So it remains to choose λ so that $a, b < 0$. For $\lambda \geq (2\sigma/\gamma)\sqrt{2n \log(ep/\delta)}$, we can see that

$$a \leq 2\lambda - 4\lambda < 0, \quad b \leq \gamma/2 - \gamma < 0,$$

so (34), (35) are verified—and hence lasso estimator recovers the correct support and signs—with probability at least $1 - 2\delta$

8.1 A note on the conditions

As we moved from the slow rates, to fast rates, to support recovery, the assumptions we used just got stronger and stronger. For the slow rates, we essentially assumed nothing about the predictor matrix X except for column normalization. For the fast rates, we had to additionally assume a compatibility or restricted eigenvalue condition, which roughly speaking, limited the correlations of the predictor variables (particularly concentrated over the underlying support S). For support recovery, we still needed whole lot more. The minimum eigenvalue condition on $(1/n)(X_S^T X_S)^{-1}$ is somewhat like the restricted eigenvalue condition on X . But the mutual incoherence condition is even stronger; it requires the regression coefficients

$$\eta_j(S) = (X_S^T X_S)^{-1} X_S^T X_j,$$

given by regressing each X_j on the truly active variables X_S , to be small (in ℓ_1 norm) for all $j \notin S$. In other words, no truly inactive variables can be highly correlated (or well-explained, in a linear projection sense) by any of the truly active variables. Finally, this minimum signal condition ensures that the nonzero entries of the true coefficient vector β_0 are big enough to detect. This is quite restrictive and is not needed for risk bounds, but it is crucial to support recovery.

8.2 Minimax bounds

Under the data model (??) with X fixed, subject to the scaling $\|X_j\|_2^2 \leq n$, for $j = 1, \dots, p$, and $\epsilon \sim N(0, \sigma^2)$, [Raskutti et al. \(2011\)](#) derive upper and lower bounds on the minimax prediction error

$$M(s_0, n, p) = \inf_{\hat{\beta}} \sup_{\|\beta_0\|_0 \leq s_0} \frac{1}{n} \|X\hat{\beta} - X\beta_0\|_2^2.$$

(Their analysis is actually considerably more broad than this and covers the coefficient error $\|\hat{\beta} - \beta_0\|_2$, as well ℓ_q constraints on β_0 , for $q \in [0, 1]$.) They prove that, under no additional assumptions on X ,

$$M(s_0, n, p) \lesssim \frac{s_0 \log(p/s_0)}{n},$$

with probability tending to 1

They also prove that, under a type of restricted eigenvalue condition in which

$$c_0 \leq \frac{(1/n)\|Xv\|_2^2}{\|v\|_2^2} \leq c_1 \text{ for all } v \in \mathbb{R}^p \text{ such that } \|v\|_0 \leq 2s_0,$$

for some constants $c_0 > 0$ and $c_1 < \infty$, it holds that

$$M(s_0, n, p) \gtrsim \frac{s_0 \log(p/s_0)}{n},$$

with probability at least 1/2

The implication is that, for some X , minimax optimal prediction may be able to be performed at a faster rate than $s_0 \log(p/s_0)/n$; but for low correlations, this is the rate we should expect. (This is consistent with the worst-case- X analysis of [Foster & George \(1994\)](#), who actually show the worst-case behavior is attained in the orthogonal X case)



Sparse additive models

Pradeep Ravikumar,

University of California, Berkeley, USA

and John Lafferty, Han Liu and Larry Wasserman

Carnegie Mellon University, Pittsburgh, USA

[Received April 2008. Final revision March 2009]

Summary. We present a new class of methods for high dimensional non-parametric regression and classification called sparse additive models. Our methods combine ideas from sparse linear modelling and additive non-parametric regression. We derive an algorithm for fitting the models that is practical and effective even when the number of covariates is larger than the sample size. Sparse additive models are essentially a functional version of the grouped lasso of Yuan and Lin. They are also closely related to the COSSO model of Lin and Zhang but decouple smoothing and sparsity, enabling the use of arbitrary non-parametric smoothers. We give an analysis of the theoretical properties of sparse additive models and present empirical results on synthetic and real data, showing that they can be effective in fitting sparse non-parametric models in high dimensional data.

Keywords: Additive models; Lasso; Non-parametric regression; Sparsity

1. Introduction

Substantial progress has been made recently on the problem of fitting high dimensional linear regression models of the form $Y_i = X_i^T \beta + \varepsilon_i$, for $i = 1, \dots, n$. Here Y_i is a real-valued response, X_i is a predictor and ε_i is a mean 0 error term. Finding an estimate of β when $p > n$ that is both statistically well behaved and computationally efficient has proved challenging; however, under the assumption that the vector β is sparse, the lasso estimator (Tibshirani, 1996) has been remarkably successful. The lasso estimator $\hat{\beta}$ minimizes the l_1 -penalized sum of squares

$$\sum_i (Y_i - X_i^T \beta)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

with the l_1 -penalty $\|\beta\|_1$ encouraging sparse solutions, where many components $\hat{\beta}_j$ are 0. The good empirical success of this estimator has been recently backed up by results confirming that it has strong theoretical properties; see Bunea *et al.* (2007), Greenshtein and Ritov (2004), Zhao and Yu (2007), Meinshausen and Yu (2006) and Wainwright (2006).

The non-parametric regression model $Y_i = m(X_i) + \varepsilon_i$, where m is a general smooth function, relaxes the strong assumptions that are made by a linear model but is much more challenging in high dimensions. Hastie and Tibshirani (1999) introduced the class of additive models of the form

Address for correspondence: Larry Wasserman, Department of Statistics, 232 Baker Hall, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA.
E-mail: larry@stat.cmu.edu

$$Y_i = \sum_{j=1}^p f_j(X_{ij}) + \varepsilon_i. \quad (1)$$

This additive combination of univariate functions—one for each covariate X_j —is less general than joint multivariate non-parametric models but can be more interpretable and easier to fit; in particular, an additive model can be estimated by using a co-ordinate descent Gauss–Seidel procedure, called backfitting. Unfortunately, additive models only have good statistical and computational behaviour when the number of variables p is not large relative to the sample size n , so their usefulness is limited in the high dimensional setting.

In this paper we investigate sparse additive models (SPAMs), which extend the advantages of sparse linear models to the additive non-parametric setting. The underlying model is the same as in equation (1), but we impose a sparsity constraint on the index set $\{j : f_j \not\equiv 0\}$ of functions f_j that are not identically zero. Lin and Zhang (2006) have proposed COSSO, an extension of the lasso to this setting, for the case where the component functions f_j belong to a reproducing kernel Hilbert space. They penalized the sum of the reproducing kernel Hilbert space norms of the component functions. Yuan (2007) proposed an extension of the non-negative garotte to this setting. As with the parametric non-negative garotte, the success of this method depends on the initial estimates of component functions f_j .

In Section 3, we formulate an optimization problem in the population setting that induces sparsity. Then we derive a sample version of the solution. The SPAM estimation procedure that we introduce allows the use of arbitrary non-parametric smoothing techniques, effectively resulting in a combination of the lasso and backfitting. The algorithm extends to classification problems by using generalized additive models. As we explain later, SPAMs can also be thought of as a functional version of the grouped lasso (Antoniadis and Fan, 2001; Yuan and Lin, 2006).

The main results of this paper include the formulation of a convex optimization problem for estimating a SPAM, an efficient backfitting algorithm for constructing the estimator and theoretical results that analyse the effectiveness of the estimator in the high dimensional setting. Our theoretical results are of two different types. First, we show that, under suitable choices of the design parameters, the SPAM backfitting algorithm recovers the correct sparsity pattern asymptotically; this is a property that we call *sparsistency*, as a shorthand for ‘sparsity pattern consistency’. Second, we show that the estimator is *persistent*, in the sense of Greenshtein and Ritov (2004), which is a form of risk consistency.

In the following section we establish notation and assumptions. In Section 3 we formulate SPAMs as an optimization problem and derive a scalable backfitting algorithm. Examples showing the use of our sparse backfitting estimator on high dimensional data are included in Section 5. In Section 6.1 we formulate the sparsistency result, when orthogonal function regression is used for smoothing. In Section 6.2 we give the persistence result. Section 7 contains a discussion of the results and possible extensions. Proofs are contained in Appendix A.

The statements of the theorems in this paper were given, without proof, in Ravikumar *et al.* (2008). The backfitting algorithm was also presented there. Related results were obtained in Meier *et al.* (2008) and Koltchinskii and Yuan (2008).

2. Notation and assumptions

We assume that we are given independent data $(X_1, Y_1), \dots, (X_n, Y_n)$ where $X_i = (X_{i1}, \dots, X_{ip})^\top \in [0, 1]^p$ and

$$Y_i = m(X_i) + \varepsilon_i \quad (2)$$

with $\varepsilon_i \sim N(0, \sigma^2)$ independent of X_i and

$$m(x) = \sum_{j=1}^p f_j(x_j). \quad (3)$$

Let μ denote the distribution of X , and let μ_j denote the marginal distribution of X_j for each $j = 1, \dots, p$. For a function f_j on $[0, 1]$ denote its $L_2(\mu_j)$ norm by

$$\|f_j\|_{\mu_j} = \sqrt{\left\{ \int_0^1 f_j^2(x) d\mu_j(x) \right\}} = \sqrt{\mathbb{E}\{f_j(X_j)^2\}}. \quad (4)$$

When the variable X_j is clear from the context, we remove the dependence on μ_j in the notation $\|\cdot\|_{\mu_j}$ and simply write $\|f_j\|$.

For $j \in \{1, \dots, p\}$, let \mathcal{H}_j denote the Hilbert subspace $L_2(\mu_j)$ of measurable functions $f_j(x_j)$ of the single scalar variable x_j with zero mean, $\mathbb{E}\{f_j(X_j)\} = 0$. Thus, \mathcal{H}_j has the inner product

$$\langle f_j, f'_j \rangle = \mathbb{E}\{f_j(X_j) f'_j(X_j)\} \quad (5)$$

and $\|f_j\| = \sqrt{\mathbb{E}\{f_j(X_j)^2\}} < \infty$. Let $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \dots \oplus \mathcal{H}_p$ denote the Hilbert space of functions of (x_1, \dots, x_p) that have the additive form: $m(x) = \sum_j f_j(x_j)$, with $f_j \in \mathcal{H}_j$, $j = 1, \dots, p$.

Let $\{\psi_{jk}, k = 0, 1, \dots\}$ denote a uniformly bounded, orthonormal basis with respect to $L^2[0, 1]$. Unless stated otherwise, we assume that $f_j \in \mathcal{T}_j$ where

$$\mathcal{T}_j = \left\{ f_j \in \mathcal{H}_j : f_j(x_j) = \sum_{k=0}^{\infty} \beta_{jk} \psi_{jk}(x_j), \sum_{k=0}^{\infty} \beta_{jk}^2 k^{2\nu_j} \leq C^2 \right\} \quad (6)$$

for some $0 < C < \infty$. We shall take $\nu_j = 2$ although the extension to other levels of smoothness is straightforward. It is also possible to adapt to ν_j although we do not pursue that direction here.

Let $\Lambda_{\min}(A)$ and $\Lambda_{\max}(A)$ denote the minimum and maximum eigenvalues of a square matrix A . If $v = (v_1, \dots, v_k)^T$ is a vector, we use the norms

$$\|v\| = \sqrt{\left(\sum_{j=1}^k v_j^2 \right)}, \quad \|v\|_1 = \sum_{j=1}^k |v_j|, \quad \|v\|_\infty = \max_j |v_j|. \quad (7)$$

3. Sparse backfitting

The outline of the derivation of our algorithm is as follows. We first formulate a population level optimization problem and show that the minimizing functions can be obtained by iterating through a series of soft thresholded univariate conditional expectations. We then plug in smoothed estimates of these univariate conditional expectations, to derive our sparse backfitting algorithm.

3.1. Population sparse additive models

For simplicity, assume that $\mathbb{E}(Y_i) = 0$. The standard additive model optimization problem in $L_2(\mu)$ (the population setting) is

$$\min_{f_j \in \mathcal{H}_j, 1 \leq j \leq p} \left[\mathbb{E} \left\{ Y - \sum_{j=1}^p f_j(X_j) \right\}^2 \right] \quad (8)$$

where the expectation is taken with respect to X and the noise ε . Now consider the following modification of this problem that introduces a scaling parameter for each function, and that imposes additional constraints:

$$\min_{\beta \in \mathbb{R}^p, g_j \in \mathcal{H}_j} \left[\mathbb{E} \left\{ Y - \sum_{j=1}^p \beta_j g_j(X_j) \right\}^2 \right] \quad (9)$$

subject to

$$\sum_{j=1}^p |\beta_j| \leq L, \quad (10)$$

$$\mathbb{E}(g_j^2) = 1, \quad j = 1, \dots, p, \quad (11)$$

noting that g_j is a function whereas $\beta = (\beta_1, \dots, \beta_p)^T$ is a vector. The constraint that β lies in the l_1 -ball $\{\beta : \|\beta\|_1 \leq L\}$ encourages sparsity of the estimated β , just as for the parametric lasso (Tibshirani, 1996). It is convenient to absorb the scaling constants β_j into the functions f_j , and to re-express the minimization in the following equivalent Lagrangian form:

$$\mathcal{L}(f, \lambda) = \frac{1}{2} \mathbb{E} \left\{ Y - \sum_{j=1}^p f_j(X_j) \right\}^2 + \lambda \sum_{j=1}^p \sqrt{\mathbb{E}\{f_j^2(X_j)\}}. \quad (12)$$

Theorem 1. The minimizers $f_j \in \mathcal{H}_j$ of equation (12) satisfy

$$f_j = \left[1 - \frac{\lambda}{\sqrt{\mathbb{E}(P_j^2)}} \right]_+ P_j \quad \text{almost surely} \quad (13)$$

where $[\cdot]_+$ denotes the positive part, and $P_j = \mathbb{E}(R_j | X_j)$ denotes the projection of the residual $R_j = Y - \sum_{k \neq j} f_k(X_k)$ onto \mathcal{H}_j .

An outline of the proof of this theorem appears in Ravikumar *et al.* (2008). A formal proof is given in Appendix A. At the population level, the f_j s can be found by a co-ordinate descent procedure that fixes $(f_k : k \neq j)$ and fits f_j by equation (13), and then iterates over j .

3.2. Data version of sparse additive models

To obtain a sample version of the population solution, we insert sample estimates into the population algorithm, as in standard backfitting (Hastie and Tibshirani, 1999). Thus, we estimate the projection $P_j = \mathbb{E}(R_j | X_j)$ by smoothing the residuals:

$$\hat{P}_j = \mathcal{S}_j R_j \quad (14)$$

where \mathcal{S}_j is a linear smoother, such as a local linear or kernel smoother. Let

$$\hat{s}_j = \frac{1}{\sqrt{n}} \|\hat{P}_j\| = \sqrt{\text{mean}(\hat{P}_j^2)} \quad (15)$$

be the estimate of $\sqrt{\mathbb{E}(P_j^2)}$. Using these plug-in estimates in the co-ordinate descent procedure yields the SPAM backfitting algorithm that is given in Table 1.

This algorithm can be seen as a functional version of the co-ordinate descent algorithm for solving the lasso. In particular, if we solve the lasso by iteratively minimizing with respect to a single co-ordinate, each iteration is given by soft thresholding; Table 2. Convergence properties of variants of this simple algorithm have been recently treated by Daubechies *et al.* (2004, 2007). Our sparse backfitting algorithm is a direct generalization of this algorithm, and it reduces to it in the case where the smoothers are local linear smoothers with large bandwidths, i.e., as the bandwidth approaches ∞ , the local linear smoother approaches a global linear fit, yielding the estimator $\hat{P}_j(i) = \hat{\beta}_j X_{ij}$. When the variables are standardized,

Table 1. SPAM backfitting algorithm†

<p><i>Input:</i> data (X_i, Y_i), regularization parameter λ <i>Initialize</i> $\hat{f}_j = 0$, for $j = 1, \dots, p$ <i>Iterate until convergence, for each</i> $j = 1, \dots, p$</p> <p>Step 1: compute the residual, $R_j = Y - \sum_{k \neq j} \hat{f}_k(X_k)$ Step 2: estimate $P_j = \mathbb{E}(R_j X_j)$ by smoothing, $\hat{P}_j = S_j R_j$ Step 3: estimate the norm, $\hat{s}_j = (1/n) \sum_{i=1}^n \hat{P}_j^2(i)$ Step 4: soft threshold, $\hat{f}_j = [1 - \lambda/\hat{s}_j]_+ \hat{P}_j$ Step 5: centre, $\hat{f}_j \leftarrow \hat{f}_j - \text{mean}(\hat{f}_j)$</p> <p><i>Output:</i> component functions \hat{f}_j and estimator $\hat{m}(X_i) = \sum_j \hat{f}_j(X_{ij})$</p>

†The first two steps in the iterative algorithm are the usual backfitting procedure; the remaining steps carry out functional soft thresholding.

Table 2. Co-ordinate descent lasso†

<p><i>Input:</i> data (X_i, Y_i), regularization parameter λ <i>Initialize</i> $\hat{\beta}_j = 0$, for $j = 1, \dots, p$ <i>Iterate until convergence, for each</i> $j = 1, \dots, p$</p> <p>Step 1: compute the residual, $R_j = Y - \sum_{k \neq j} \hat{\beta}_k X_k$ Step 2: project residual onto X_j, $P_j = X_j^T R_j$ Step 3: soft threshold, $\hat{\beta}_j = [1 - \lambda/ P_j]_+ P_j$</p> <p><i>Output:</i> estimator $\hat{m}(X_i) = \sum_j \hat{\beta}_j X_{ij}$</p>

†The SPAM backfitting algorithm is a functional version of the co-ordinate descent algorithm for the lasso, which computes $\hat{\beta} = \arg \min(\frac{1}{2} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1)$.

$$\hat{s}_j = \sqrt{\left(\frac{1}{n} \sum_{i=1}^n \hat{\beta}_j^2 X_{ij}^2 \right)} = |\hat{\beta}_j|$$

so the soft thresholding in step 4 of the SPAM backfitting algorithm is the same as the soft thresholding in step 3 in the co-ordinate descent lasso algorithm.

3.3. Basis functions

It is useful to express the model in terms of basis functions. Recall that $B_j = (\psi_{jk} : k = 1, 2, \dots)$ is an orthonormal basis for T_j and that $\sup_x |\psi_{jk}(x)| \leq B$ for some B . Then

$$f_j(x_j) = \sum_{k=1}^{\infty} \beta_{jk} \psi_{jk}(x_j) \quad (16)$$

where $\beta_{jk} = \int f_j(x_j) \psi_{jk}(x_j) dx_j$.

Let us also define

$$\tilde{f}_j(x_j) = \sum_{k=1}^d \beta_{jk} \psi_{jk}(x_j) \quad (17)$$

where $d = d_n$ is a truncation parameter. For the Sobolev space T_j of order 2 we have that $\|f_j - \tilde{f}_j\|^2 = O(1/d^4)$. Let $S = \{j : f_j \neq 0\}$. Assuming the sparsity condition $|S| = O(1)$ it follows that $\|m - \tilde{m}\|^2 = O(1/d^4)$ where $\tilde{m} = \sum_j \tilde{f}_j$. The usual choice is $d \asymp n^{1/5}$, yielding truncation bias $\|m - \tilde{m}\|^2 = O(n^{-4/5})$.

In this setting, the smoother can be taken to be the least squares projection onto the truncated set of basis functions $\{\psi_{j1}, \dots, \psi_{jd}\}$; this is also called orthogonal series smoothing. Let Ψ_j denote the $n \times d_n$ matrix that is given by $\Psi_j(i, l) = \psi_{j,l}(X_{ij})$. The smoothing matrix is the projection matrix $S_j = \Psi_j(\Psi_j^T \Psi_j)^{-1} \Psi_j^T$. In this case, the backfitting algorithm in Table 1 is a co-ordinate descent algorithm for minimizing

$$\frac{1}{2n} \left\| Y - \sum_{j=1}^p \Psi_j \beta_j \right\|_2^2 + \lambda \sum_{j=1}^p \sqrt{\left(\frac{1}{n} \beta_j^T \Psi_j^T \Psi_j \beta_j \right)}$$

which is the sample version of equation (12). This is the Lagrangian of a second-order cone program, and standard convexity theory implies the existence of a minimizer. In Section 6.1 we prove theoretical properties of SPAMs by assuming that this particular smoother is being used.

3.4. Connection with the grouped lasso

The SPAM model can be thought of as a functional version of the grouped lasso (Yuan and Lin, 2006) as we now explain. Consider the following linear regression model with multiple factors:

$$Y = \sum_{j=1}^{p_n} X_j \beta_j + \varepsilon = X \beta + \varepsilon, \quad (18)$$

where Y is an $n \times 1$ response vector, ε is an $n \times 1$ vector of independent and identically distributed mean 0 noise, X_j is an $n \times d_j$ matrix corresponding to the j th factor and β_j is the corresponding $d_j \times 1$ coefficient vector. Assume for convenience (in this subsection only) that each X_j is orthogonal, so that $X_j^T X_j = I_{d_j}$, where I_{d_j} is the $d_j \times d_j$ identity matrix. We use $X = (X_1, \dots, X_{p_n})$ to denote the full design matrix and use $\beta = (\beta_1^T, \dots, \beta_{p_n}^T)^T$ to denote the parameter.

The *grouped lasso* estimator is defined as the solution of the following convex optimization problem:

$$\hat{\beta}(\lambda_n) = \arg \min_{\beta} \left(\|Y - X\beta\|_2^2 + \lambda_n \sum_{j=1}^{p_n} \sqrt{d_j} \|\beta_j\| \right) \quad (19)$$

where $\sqrt{d_j}$ scales the j th term to compensate for different group sizes.

It is obvious that, when $d_j = 1$ for $j = 1, \dots, p_n$, the grouped lasso becomes the standard lasso. From the Karush–Kuhn–Tucker optimality conditions, a necessary and sufficient condition for $\hat{\beta} = (\hat{\beta}_1^T, \dots, \hat{\beta}_{p_n}^T)^T$ to be the grouped lasso solution is

$$\begin{aligned} -X_j^T(Y - X\hat{\beta}) + \frac{\lambda \sqrt{d_j} \hat{\beta}_j}{\|\hat{\beta}_j\|} &= \mathbf{0}, & \forall \hat{\beta}_j \neq \mathbf{0}, \\ \|X_j^T(Y - X\hat{\beta})\| &\leq \lambda \sqrt{d_j}, & \forall \hat{\beta}_j = \mathbf{0}. \end{aligned} \quad (20)$$

On the basis of this stationary condition, an iterative blockwise co-ordinate descent algorithm can be derived; as shown by Yuan and Lin (2006), a solution to equation (20) satisfies

$$\hat{\beta}_j = \left[1 - \frac{\lambda \sqrt{d_j}}{\|S_j\|} \right]_+ S_j \quad (21)$$

where $S_j = X_j^T(Y - X\beta_{\setminus j})$, with $\beta_{\setminus j} = (\beta_1^T, \dots, \beta_{j-1}^T, \mathbf{0}^T, \beta_{j+1}^T, \dots, \beta_{p_n}^T)$. By iteratively applying equation (21), the grouped lasso solution can be obtained.

As discussed in Section 1, the COSSO model of Lin and Zhang (2006) replaces the lasso constraint on $\sum_j |\beta_j|$ with a reproducing kernel Hilbert space constraint. The advantage of our formulation is that it decouples smoothness ($g_j \in \mathcal{T}_j$) and sparsity ($\sum_j |\beta_j| \leq L$). This leads to a

simple algorithm that can be carried out with any non-parametric smoother and scales easily to high dimensions.

4. Choosing the regularization parameter

We choose λ by minimizing an estimate of the risk. Let ν_j be the effective degrees of freedom for the smoother on the j th variable, i.e. $\nu_j = \text{tr}(\mathcal{S}_j)$ where \mathcal{S}_j is the smoothing matrix for the j th dimension. Also let $\hat{\sigma}^2$ be an estimate of the variance. Define the total effective degrees of freedom as

$$\text{df}(\lambda) = \sum_j \nu_j I(\|\hat{f}_j\| \neq 0). \quad (22)$$

Two estimates of risk are

$$C_p = \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \sum_{j=1}^p \hat{f}_j(X_{ij}) \right\}^2 + \frac{2\hat{\sigma}^2}{n} \text{df}(\lambda) \quad (23)$$

and

$$\text{GCV}(\lambda) = \frac{(1/n) \sum_{i=1}^n \left\{ Y_i - \sum_j \hat{f}_j(X_{ij}) \right\}^2}{\{1 - \text{df}(\lambda)/n\}^2}. \quad (24)$$

The first is C_p and the second is generalized cross-validation but with degrees of freedom defined by $\text{df}(\lambda)$. A proof that these are valid estimates of risk is not currently available; thus, these should be regarded as heuristics.

On the basis of the results in Wasserman and Roeder (2007) about the lasso, it seems likely that choosing λ by risk estimation can lead to overfitting. One can further clean the estimate by testing $H_0: f_j = 0$ for all j such that $\hat{f}_j \neq 0$. For example, the tests in Fan and Jiang (2005) could be used.

5. Examples

To illustrate the method, we consider a few examples.

5.1. Synthetic data

We generated $n = 100$ observations for an additive model with $p = 100$ and four relevant variables,

$$Y_i = \sum_{j=1}^4 f_j(X_{ij}) + \varepsilon_i,$$

where $\varepsilon_i \sim N(0, 1)$; the relevant component functions are given by

$$\begin{aligned} f_1(x) &= -\sin(1.5x), \\ f_2(x) &= x^3 + 1.5(x - 0.5)^2, \\ f_3(x) &= -\phi(x, 0.5, 0.8^2), \\ f_4(x) &= \sin\{\exp(-0.5x)\} \end{aligned}$$

where $\phi(\cdot, 0.5, 0.8^2)$ is the Gaussian probability distribution function with mean 0.5 and standard deviation 0.8. The data therefore have 96 irrelevant dimensions. The covariates are sampled independent and identically distributed from uniform($-2.5, 2.5$). All the component functions are standardized, i.e.

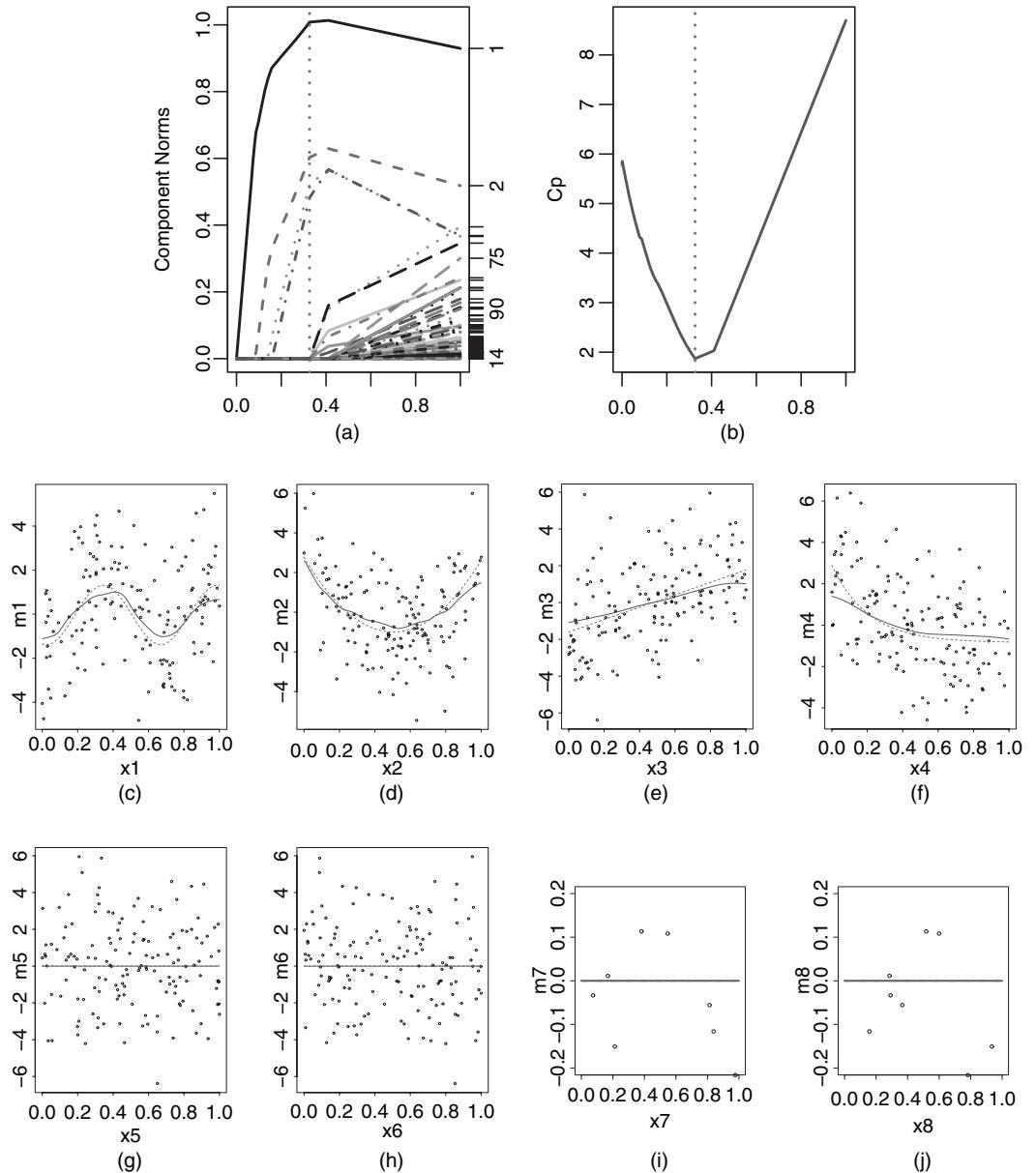


Fig. 1. Simulated data: (a) empirical ℓ_2 -norm of the estimated components as plotted against the regularization parameter λ (the value on the x -axis is proportional to $\sum_j \|\hat{\beta}_j\|_1$); (b) C_p -scores against the amount of regularization (\cdot , value of λ which has the smallest C_p -score); estimated (—) versus true additive component functions (---) for (c)–(f) the first four relevant dimensions and (g)–(j) the first four irrelevant dimensions ((c) $\ell_1 = 97.05$; (d) $\ell_1 = 88.36$; (e) $\ell_1 = 90.65$; (f) $\ell_1 = 79.26$; (g)–(j) $\ell_1 = 0$)

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n f_j(X_{ij}) &= 0, \\ \frac{1}{n-1} \sum_{i=1}^n f_j^2(X_{ij}) &= 1. \end{aligned} \quad (25)$$

The results of applying SPAMs are summarized in Fig. 1, using the plug-in bandwidths

$$h_j = 0.6 \text{sd}(X_j)/n^{1/5}.$$

Fig. 1(a) shows regularization paths as the parameter λ varies; each curve is a plot of $\|\hat{f}_j(\lambda)\|$ versus

$$\sum_{k=1}^p \|\hat{f}_k(\lambda)\| / \max_{\lambda} \left\{ \sum_{k=1}^p \|\hat{f}_k(\lambda)\| \right\} \quad (26)$$

for a particular variable X_j . The estimates are generated efficiently over a sequence of λ -values by ‘warm starting’ $\hat{f}_j(\lambda_t)$ at the previous value $\hat{f}_j(\lambda_{t-1})$. Fig. 1(b) shows the C_p -statistic as a function of regularization level.

5.2. Functional sparse coding

Olshausen and Field (1996) proposed a method of obtaining sparse representations of data such as natural images; the motivation comes from trying to understand principles of neural coding. In this example we suggest a non-parametric form of sparse coding.

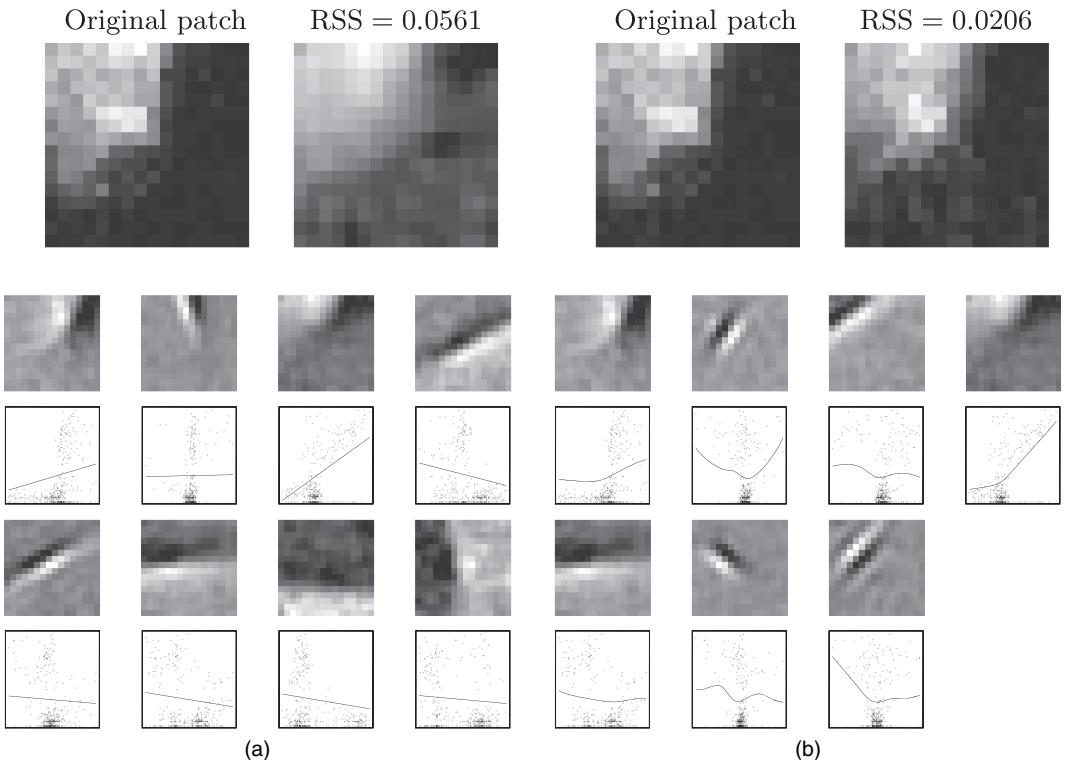


Fig. 2. Comparison of sparse reconstructions by using (a) the lasso and (b) SPAMs

Let $\{y^i\}_{i=1,\dots,N}$ be the data to be represented with respect to some learned basis, where each instance $y^i \in \mathbb{R}^n$ is an n -dimensional vector. The linear sparse coding optimization problem is

$$\min_{\beta, X} \left\{ \sum_{i=1}^N \left(\frac{1}{2n} \|y^i - X\beta^i\|^2 + \lambda \|\beta^i\|_1 \right) \right\} \quad (27)$$

such that

$$\|X_j\| \leq 1. \quad (28)$$

Here X is an $n \times p$ matrix with columns X_j , representing the ‘dictionary’ entries or basis vectors to be learned. It is not required that the basis vectors are orthogonal. The l_1 -penalty on the coefficients β^i encourages sparsity, so each data vector y^i is represented by only a small number of dictionary elements. Sparsity allows the features to specialize, and to capture salient properties of the data.

This optimization problem is not jointly convex in β^i and X . However, for fixed X , each weight vector β^i is computed by running the lasso. For fixed β^i , the optimization is similar to ridge regression and can be solved efficiently. Thus, an iterative procedure for (approximately) solving this optimization problem is easy to derive.

In the case of sparse coding of natural images, as in Olshausen and Field (1996), the basis vectors X_j encode basic edge features at different scales and spatial orientations. In the functional version, we no longer assume a linear parametric fit between the dictionary X and the data y . Instead, we model the relationship by using an additive model. This leads to the following optimization problem for functional sparse coding:

$$\min_{f, X} \left[\sum_{i=1}^N \left\{ \frac{1}{2n} \left\| y^i - \sum_{j=1}^p f_j^i(X_j) \right\|^2 + \lambda \sum_{j=1}^p \|f_j^i\| \right\} \right] \quad (29)$$

such that

$$\|X_j\| \leq 1, \quad j = 1, \dots, p. \quad (30)$$

Fig. 2 illustrates the reconstruction of various image patches by using the sparse linear model compared with the SPAM. Local linear smoothing was used with a Gaussian kernel having fixed bandwidth $h = 0.05$ for all patches and all codewords. The codewords X_j are those obtained by using the Olshausen-Field procedure; these become the design points in the regression estimators. Thus, a codeword for a 16×16 patch corresponds to a vector X_j of dimension 256, with each X_{ij} the grey level for a particular pixel.

6. Theoretical properties

6.1. Sparsistency

In the case of linear regression, with $f_j(X_j) = \beta_j^{*\top} X_j$, several researchers have shown that, under certain conditions on n and p , the number of relevant variables $s = |\text{supp}(\beta^*)|$, and the design matrix X , the lasso recovers the sparsity pattern asymptotically, i.e. the lasso estimator $\hat{\beta}_n$ is *sparsistent*:

$$\mathbb{P}\{\text{supp}(\beta^*) = \text{supp}(\hat{\beta}_n)\} \rightarrow 1. \quad (31)$$

Here, $\text{supp}(\beta) = \{j : \beta_j \neq 0\}$. References include Wainwright (2006), Meinshausen and Bühlmann (2006), Zou (2005), Fan and Li (2001) and Zhao and Yu (2007). We show a similar result for SPAMs under orthogonal function regression.

In terms of an orthogonal basis ψ , we can write

$$Y_i = \sum_{j=1}^p \sum_{k=1}^{\infty} \beta_{jk}^* \psi_{jk}(X_{ij}) + \varepsilon_i. \quad (32)$$

To simplify the notation, let β_j be the d_n -dimensional vector $\{\beta_{jk}, k = 1, \dots, d_n\}$ and let Ψ_j be the $n \times d_n$ matrix $\Psi_j(i, k) = \psi_{jk}(X_{ij})$. If $A \subset \{1, \dots, p\}$, we denote by Ψ_A the $n \times d|A|$ matrix where, for each $j \in A$, Ψ_j appears as a submatrix in the natural way.

We now analyse the sparse backfitting algorithm of Table 1 by assuming that an orthogonal series smoother is used to estimate the conditional expectation in its step 2. As noted earlier, an orthogonal series smoother for a predictor X_j is the least squares projection onto a truncated set of basis functions $\{\psi_{j1}, \dots, \psi_{jd}\}$. Our optimization problem in this setting is

$$\min_{\beta} \left\{ \frac{1}{2n} \left\| Y - \sum_{j=1}^p \Psi_j \beta_j \right\|_2^2 + \lambda \sum_{j=1}^p \sqrt{\left(\frac{1}{n} \beta_j^T \Psi_j^T \Psi_j \beta_j \right)} \right\}. \quad (33)$$

Combined with the soft thresholding step, the update for f_j in the algorithm in Table 1 can thus be seen to solve the problem

$$\min_{\beta} \left\{ \frac{1}{2n} \|R_j - \Psi_j \beta_j\|_2^2 + \lambda_n \sqrt{\left(\frac{1}{n} \beta_j^T \Psi_j^T \Psi_j \beta_j \right)} \right\}$$

where $\|v\|_2^2$ denotes $\sum_{i=1}^n v_i^2$ and $R_j = Y - \sum_{l \neq j} \Psi_l \beta_l$ is the residual for f_j . The sparse backfitting algorithm thus solves

$$\min_{\beta} \{R_n(\beta) + \lambda_n \Omega(\beta)\} = \min_{\beta} \left(\frac{1}{2n} \left\| Y - \sum_{j=1}^p \Psi_j \beta_j \right\|_2^2 + \lambda_n \sum_{j=1}^p \left\| \frac{1}{\sqrt{n}} \Psi_j \beta_j \right\|_2 \right) \quad (34)$$

where R_n denotes the squared error term and Ω denotes the regularization term, and each β_j is a d_n -dimensional vector. Let S denote the true set of variables $\{j : f_j \neq 0\}$, with $s = |S|$, and let S^c denote its complement. Let $\hat{S}_n = \{j : \hat{\beta}_j \neq 0\}$ denote the estimated set of variables from the minimizer $\hat{\beta}_n$, with corresponding function estimates $\hat{f}_j(x_j) = \sum_{k=1}^{d_n} \hat{\beta}_{jk} \psi_{jk}(x_j)$. For the results in this section, we shall treat the covariates as fixed. A preliminary version of the following result is stated, without proof, in Ravikumar *et al.* (2008).

Theorem 2. Suppose that the following conditions hold on the design matrix X in the orthogonal basis ψ :

$$\Lambda_{\max} \left(\frac{1}{n} \Psi_S^T \Psi_S \right) \leq C_{\max} < \infty, \quad (35)$$

$$\Lambda_{\min} \left(\frac{1}{n} \Psi_S^T \Psi_S \right) \geq C_{\min} > 0, \quad (36)$$

$$\max_{j \in S^c} \left\| \left(\frac{1}{n} \Psi_j^T \Psi_S \right) \left(\frac{1}{n} \Psi_S^T \Psi_S \right)^{-1} \right\| \leq \sqrt{\left(\frac{C_{\min}}{C_{\max}} \right) \frac{1-\delta}{\sqrt{s}}}, \quad \text{for some } 0 < \delta \leq 1. \quad (37)$$

Assume that the truncation dimension d_n satisfies $d_n \rightarrow \infty$ and $d_n = o(n)$. Furthermore, suppose the following conditions, which relate the regularization parameter λ_n to the design parameters n and p , the number of relevant variables s and the truncation size d_n :

$$\frac{s}{d_n \lambda_n} \rightarrow 0, \quad (38)$$

$$\frac{d_n \log\{d_n(p-s)\}}{n\lambda_n^2} \rightarrow 0, \quad (39)$$

$$\frac{1}{\rho_n^*} \left[\sqrt{\left\{ \frac{\log(sd_n)}{n} \right\}} + \frac{s^{3/2}}{d_n} + \lambda_n \sqrt{(sd_n)} \right] \rightarrow 0 \quad (40)$$

where $\rho_n^* = \min_{j \in S} \|\beta_j^*\|_\infty$. Then the solution $\hat{\beta}_n$ to problem (33) is unique and satisfies $\hat{S}_n = S$ with probability approaching 1.

This result parallels the theorem of Wainwright (2006) on model selection consistency of the lasso; however, technical subtleties arise because of the truncation dimension d_n which is increasing with sample size, and the matrix $\Psi_j^T \Psi$ which appears in the regularization of β_j . As a result, the operator norm rather than the ∞ -norm appears in the incoherence condition (37). Note, however, that condition (37) implies that

$$\|\Psi_{S^c}^T \Psi_S (\Psi_S^T \Psi_S)^{-1}\|_\infty = \max_{j \in S^c} \|\Psi_j^T \Psi_S (\Psi_S^T \Psi_S)^{-1}\|_\infty \quad (41)$$

$$\leq \sqrt{\left(\frac{C_{\min} d_n}{C_{\max}} \right)} (1 - \delta) \quad (42)$$

since $(1/\sqrt{n})\|A\|_\infty \leq \|A\| \leq \sqrt{m}\|A\|_\infty$ for an $m \times n$ matrix A . This relates it to the more standard incoherence conditions that have been used for sparsistency in the case of the lasso.

The following corollary, which imposes the additional condition that the number of relevant variables is bounded, follows directly. It makes explicit how to choose the design parameters d_n and λ_n , and implies a condition on the fastest rate at which the minimum norm ρ_n^* can approach 0.

Corollary 1. Suppose that $s = O(1)$, and assume that the design conditions (35)–(37) hold. If the truncation dimension d_n , regularization parameter λ_n and minimum norm ρ_n^* satisfy

$$d_n \asymp n^{1/3}, \quad (43)$$

$$\lambda_n \asymp \frac{\log(np)}{n^{1/3}}, \quad (44)$$

$$\frac{1}{\rho_n^*} = o\left\{ \frac{n^{1/6}}{\log(np)} \right\} \quad (45)$$

then $\mathbb{P}(\hat{S}_n = S) \rightarrow 1$.

The following proposition clarifies the implications of condition (45), by relating the sup-norm $\|\beta_j\|_\infty$ to the function norm $\|f_j\|_2$.

Proposition 1. Suppose that $f(x) = \sum_k \beta_k \psi_k(x)$ is in the Sobolev space of order $\nu > \frac{1}{2}$, so that $\sum_{i=1}^\infty \beta_i^2 i^{2\nu} \leq C^2$ for some constant C . Then

$$\|f\|_2 = \|\beta\|_2 \leq c \|\beta\|_\infty^{2\nu/(2\nu+1)} \quad (46)$$

for some constant c .

For instance, the result of corollary 1 allows the norms of the coefficients β_j to decrease as $\|\beta_j\|_\infty = \log^2(np)/n^{1/6}$. In the case $\nu = 2$, this would allow the norms $\|f_j\|_2$ of the relevant functions to approach 0 at the rate $\log^{8/5}(np)/n^{2/15}$.

6.2. Persistence

The previous assumptions are very strong. They can be weakened at the expense of obtaining weaker results. In particular, in this section we do not assume that the true regression function is additive. We use arguments like those in Juditsky and Nemirovski (2000) and Greenshteyn and Ritov (2004) in the context of linear models. In this section we treat X as random and we use triangular array asymptotics, i.e. the joint distribution for the data can change with n . Let (X, Y) denote a new pair (independent of the observed data) and define the predictive risk when predicting Y with $v(X)$ by

$$R(v) = \mathbb{E}\{Y - v(X)\}^2. \quad (47)$$

When $v(x) = \sum_j \beta_j g_j(x_j)$ we also write the risk as $R(\beta, g)$ where $\beta = (\beta_1, \dots, \beta_p)$ and $g = (g_1, \dots, g_p)$. Following Greenshteyn and Ritov (2004) we say that an estimator \hat{m}_n is persistent (risk consistent) relative to a class of functions \mathcal{M}_n , if

$$R(\hat{m}_n) - R(m_n^*) \xrightarrow{P} 0 \quad (48)$$

where

$$m_n^* = \arg \min_{v \in \mathcal{M}_n} \{R(v)\} \quad (49)$$

is the predictive oracle. Greenshteyn and Ritov (2004) showed that the lasso is persistent for $\mathcal{M}_n = \{l(x) = x^T \beta : \|\beta\|_1 \leq L_n\}$ and $L_n = o\{n/\log(n)^{1/4}\}$. Note that m_n^* is the best linear approximation (in prediction risk) in \mathcal{M}_n but the true regression function is not assumed to be linear. Here we show a similar result for SPAMs.

In this section, we assume that the SPAM estimator \hat{m}_n is chosen to minimize

$$\frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \sum_j \beta_j g_j(X_{ij}) \right\}^2 \quad (50)$$

subject to $\|\beta\|_1 \leq L_n$ and $g_j \in \mathcal{T}_j$. We make no assumptions about the design matrix. Let $\mathcal{M}_n \equiv \mathcal{M}_n(L_n)$ be defined by

$$\mathcal{M}_n = \left\{ m : m(x) = \sum_{j=1}^{p_n} \beta_j g_j(x_j) : \mathbb{E}(g_j) = 0, \mathbb{E}(g_j^2) = 1, \sum_j |\beta_j| \leq L_n \right\} \quad (51)$$

and let $m_n^* = \arg \min_{v \in \mathcal{M}_n} \{R(v)\}$.

Theorem 3. Suppose that $p_n \leq \exp(n^\xi)$ for some $\xi < 1$. Then,

$$R(\hat{m}_n) - R(m_n^*) = O_P\left(\frac{L_n^2}{n^{(1-\xi)/2}}\right) \quad (52)$$

and hence, if $L_n = o(n^{(1-\xi)/4})$, then the SPAM is persistent.

7. Discussion

The results that are presented here show how many of the recently established theoretical properties of l_1 -regularization for linear models extend to SPAMs. The sparse backfitting algorithm that we have derived is attractive because it decouples smoothing and sparsity, and can be used with any non-parametric smoother. It thus inherits the nice properties of the original backfitting procedure. However, our theoretical analyses have made use of a particular form of smoothing, using a truncated orthogonal basis. An important problem is thus to extend the theory to cover more general classes of smoothing operators. Convergence properties of the SPAM backfitting

algorithm should also be investigated; convergence of special cases of standard backfitting was studied by Buja *et al.* (1989).

An additional direction for future work is to develop procedures for automatic bandwidth selection in each dimension. We have used plug-in bandwidths and truncation dimensions d_n in our experiments and theory. It is of particular interest to develop procedures that are adaptive to different levels of smoothness in different dimensions. It would also be of interest to consider more general penalties of the form $p_\lambda(\|f_j\|)$, as in Fan and Li (2001).

Finally, we note that, although we have considered basic additive models that allow functions of individual variables, it is natural to consider interactions, as in the functional analysis-of-variance model. One challenge is to formulate suitable incoherence conditions on the functions that enable regularization-based procedures or greedy algorithms to recover the correct interaction graph. In the parametric setting, one result in this direction is Wainwright *et al.* (2007).

Acknowledgements

This research was supported in part by National Science Foundation grant CCF-0625879 and a Siebel scholarship to PR.

Appendix A: Proofs

A.1. Proof of theorem 1

Consider the minimization of the Lagrangian

$$\min_{\{f_j \in \mathcal{H}_j\}} \{\mathcal{L}(f, \lambda)\} \equiv \frac{1}{2} \mathbb{E} \left\{ Y - \sum_{j=1}^p f_j(X_j) \right\}^2 + \lambda \sum_{j=1}^p \sqrt{\mathbb{E}\{f_j(X_j)^2\}} \quad (53)$$

with respect to $f_j \in \mathcal{H}_j$, holding the other components $\{f_k, k \neq j\}$ fixed. The stationary condition is obtained by setting the Fréchet derivative to 0. Denote by $\partial_j \mathcal{L}(f, \lambda; \eta_j)$ the directional derivative with respect to f_j in the direction $\eta_j(X_j) \in \mathcal{H}_j \{ \mathbb{E}(\eta_j) = 0, \mathbb{E}(\eta_j^2) < \infty \}$. Then the stationary condition can be formulated as

$$\partial_j \mathcal{L}(f, \lambda; \eta_j) = \frac{1}{2} \mathbb{E}\{(f_j - R_j + \lambda v_j) \eta_j\} = 0 \quad (54)$$

where $R_j = Y - \sum_{k \neq j} f_k$ is the residual for f_j , and $v_j \in \mathcal{H}_j$ is an element of the subgradient $\partial \sqrt{\mathbb{E}(f_j^2)}$, satisfying $v_j = f_j / \sqrt{\mathbb{E}(f_j^2)}$ if $\mathbb{E}(f_j^2) \neq 0$ and $v_j \in \{u_j \in \mathcal{H}_j | \mathbb{E}(u_j^2) \leq 1\}$ otherwise.

Using iterated expectations, the above condition can be rewritten as

$$\mathbb{E}[\{f_j + \lambda v_j - \mathbb{E}(R_j | X_j)\} \eta_j] = 0. \quad (55)$$

But, since $f_j - \mathbb{E}(R_j | X_j) + \lambda v_j \in \mathcal{H}_j$, we can compute the derivative in the direction $\eta_j = f_j - \mathbb{E}(R_j | X_j) + \lambda v_j \in \mathcal{H}_j$, implying that

$$\mathbb{E}[\{f_j(x_j) - \mathbb{E}(R_j | X_j = x_j) + \lambda v_j(x_j)\}^2] = 0, \quad (56)$$

i.e.

$$f_j + \lambda v_j = \mathbb{E}(R_j | X_j) \quad \text{almost everywhere.} \quad (57)$$

Denote the conditional expectation $\mathbb{E}(R_j | X_j)$ —also the projection of the residual R_j onto \mathcal{H}_j —by P_j . Now, if $\mathbb{E}(f_j^2) \neq 0$, then $v_j = f_j / \sqrt{\mathbb{E}(f_j^2)}$, which from condition (57) implies

$$\sqrt{\mathbb{E}(P_j^2)} = \sqrt{\mathbb{E}\{(f_j + \lambda f_j / \sqrt{\mathbb{E}(f_j^2)})^2\}} \quad (58)$$

$$= \left\{ 1 + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}} \right\} \sqrt{\mathbb{E}(f_j^2)} \quad (59)$$

$$= \sqrt{\mathbb{E}(f_j^2)} + \lambda \quad (60)$$

$$\geq \lambda. \quad (61)$$

If $\mathbb{E}(f_j^2) = 0$, then $f_j = 0$ almost everywhere, and $\sqrt{\mathbb{E}(v_j^2)} \leq 1$. Equation (57) then implies that

$$\sqrt{\mathbb{E}(P_j^2)} \leq \lambda. \quad (62)$$

We thus obtain the equivalence

$$\sqrt{\mathbb{E}(P_j^2)} \leq \lambda \Leftrightarrow f_j = 0 \quad \text{almost everywhere.} \quad (63)$$

Rewriting equation (57) in light of result (63), we obtain

$$\begin{cases} 1 + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}} & f_j = P_j \\ & \text{if } \sqrt{\mathbb{E}(P_j^2)} > \lambda, \\ & f_j = 0 \\ & \text{otherwise.} \end{cases}$$

Using equation (60), we thus arrive at the soft thresholding update for f_j :

$$f_j = \left[1 - \frac{\lambda}{\sqrt{\mathbb{E}(P_j^2)}} \right]_+ P_j \quad (64)$$

where $[\cdot]_+$ denotes the positive part and $P_j = \mathbb{E}[R_j | X_j]$.

A.2. Proof of theorem 2

A vector $\hat{\beta} \in \mathbb{R}^{d_n p}$ is an optimum of the objective function in expression (34) if and only if there is a subgradient $\hat{g} \in \partial\Omega(\hat{\beta})$, such that

$$\frac{1}{n} \Psi^T \left(\sum_j \Psi_j \hat{\beta}_j - Y \right) + \lambda_n \hat{g} = 0. \quad (65)$$

The subdifferential $\partial\Omega(\beta)$ is the set of vectors $g \in \mathbb{R}^{pd_n}$ satisfying

$$\begin{aligned} g_j &= \frac{(1/n)\Psi_j^T \Psi_j \beta_j}{\sqrt{\{(1/n)\beta_j^T \Psi_j^T \Psi_j \beta_j\}}} && \text{if } \beta_j \neq 0, \\ g_j^T \left(\frac{1}{n} \Psi_j^T \Psi_j \right)^{-1} g_j &\leq 1 && \text{if } \beta_j = 0. \end{aligned}$$

Our argument is based on the technique of a *primal dual witness*, which has been used previously in the analysis of the lasso (Wainwright, 2006). In particular, we construct a coefficient subgradient pair $(\hat{\beta}, \hat{g})$ which satisfies $\text{supp}(\hat{\beta}) = \text{supp}(\beta^*)$ and in addition satisfies the optimality conditions for the objective (34) with high probability. Thus, when the procedure succeeds, the constructed coefficient vector $\hat{\beta}$ is equal to the solution of the convex objective (34), and \hat{g} is an optimal solution to its dual. From its construction, the support of $\hat{\beta}$ is equal to the true support $\text{supp}(\beta^*)$, from which we can conclude that the solution of the objective (34) is sparsistent. The construction of the primal dual witness proceeds as follows.

- (a) Set $\hat{\beta}_{S^c} = 0$.
- (b) Set $\hat{g}_S = \partial\Omega(\beta^*)_S$.
- (c) With these settings of $\hat{\beta}_{S^c}$ and \hat{g}_S , obtain $\hat{\beta}_S$ and \hat{g}_{S^c} from the stationary conditions in equation (65).

For the witness procedure to succeed, we must show that $(\hat{\beta}, \hat{g})$ is optimal for the objective (34), meaning that

$$\hat{\beta}_j \neq 0 \quad \text{for } j \in S, \quad (66a)$$

$$g_j^T \left(\frac{1}{n} \Psi_j^T \Psi_j \right)^{-1} g_j < 1 \quad \text{for } j \in S^c. \quad (66b)$$

For uniqueness of the solution, we require strict dual feasibility, meaning strict inequality in condition (66b). In what follows, we show that these two conditions hold with high probability.

A.2.1. Condition (66a)

Setting $\hat{\beta}_{S^c} = 0$ and

$$\hat{g}_j = \frac{(1/n)\Psi_j^\top \Psi_j \beta_j^*}{\sqrt{\{(1/n)\beta_j^{*\top} \Psi_j^\top \Psi_j \beta_j^*\}}} \quad \text{for } j \in S,$$

the stationarity condition for $\hat{\beta}_S$ is given by

$$\frac{1}{n}\Psi_S^\top (\Psi_S \hat{\beta}_S - Y) + \lambda_n \hat{g}_S = 0. \quad (67)$$

Let $V = Y - \Psi_S \beta_S^* - W$ denote the error due to finite truncation of the orthogonal basis, where $W = (\varepsilon_1, \dots, \varepsilon_n)^\top$. Then the stationarity condition (67) can be simplified as

$$\frac{1}{n}\Psi_S^\top \Psi_S (\hat{\beta}_S - \beta_S^*) - \frac{1}{n}\Psi_S^\top W - \frac{1}{n}\Psi_S^\top V + \lambda_n \hat{g}_S = 0,$$

so that

$$\hat{\beta}_S - \beta_S^* = \left(\frac{1}{n}\Psi_S^\top \Psi_S \right)^{-1} \left(\frac{1}{n}\Psi_S^\top W + \frac{1}{n}\Psi_S^\top V - \lambda_n \hat{g}_S \right), \quad (68)$$

where we have used the assumption that $(1/n)\Psi_S^\top \Psi_S$ is non-singular. Recalling our definition of the minimum function norm $\rho_n^* = \min_{j \in S} \|\beta_j^*\|_\infty > 0$, it suffices to show that $\|\hat{\beta}_S - \beta_S^*\|_\infty < \rho_n^*/2$, to ensure that

$$\text{supp}(\beta_S^*) = \text{supp}(\hat{\beta}_S) = \{j : \|\hat{\beta}_j\|_\infty \neq 0\},$$

so that condition (66a) would be satisfied. Using $\Sigma_{SS} = (1/n)(\Psi_S^\top \Psi_S)$ to simplify the notation, we have the l_∞ -bound,

$$\|\hat{\beta}_S - \beta_S^*\|_\infty \leq \underbrace{\left\| \Sigma_{SS}^{-1} \left(\frac{1}{n}\Psi_S^\top W \right) \right\|_\infty}_{T_1} + \underbrace{\left\| \Sigma_{SS}^{-1} \left(\frac{1}{n}\Psi_S^\top V \right) \right\|_\infty}_{T_2} + \lambda_n \underbrace{\|\Sigma_{SS}^{-1} \hat{g}_S\|_\infty}_{T_3}. \quad (69)$$

We now proceed to bound the quantities T_1 , T_2 and T_3 .

A.2.2. Bounding T_3

Note that, for $j \in S$,

$$1 = g_j^\top \left(\frac{1}{n}\Psi_j^\top \Psi_j \right)^{-1} g_j \geq \frac{1}{C_{\max}} \|g_j\|^2,$$

and thus $\|g_j\| \leq \sqrt{C_{\max}}$. Noting further that

$$\|g_S\|_\infty = \max_{j \in S} (\|g_j\|_\infty) \leq \max_{j \in S} (\|g_j\|_2) \leq \sqrt{C_{\max}}, \quad (70)$$

it follows that

$$T_3 := \|\Sigma_{SS}^{-1} \hat{g}_S\|_\infty \leq \sqrt{C_{\max}} \|\Sigma_{SS}^{-1}\|_\infty. \quad (71)$$

A.2.3. Bounding T_2

We proceed in two steps; we first bound $\|V\|_\infty$ and use this to bound $\|(1/n)\Psi_S^\top V\|_\infty$. Note that, as we are working over the Sobolev spaces \mathcal{S}_j of order 2,

$$\begin{aligned} |V_i| &= \left| \sum_{j \in S} \sum_{k=d_n+1}^{\infty} \beta_{jk}^* \Psi_{jk}(X_{ij}) \right| \leq B \sum_{j \in S} \sum_{k=d_n+1}^{\infty} |\beta_{jk}^*| \\ &= B \sum_{j \in S} \sum_{k=d_n+1}^{\infty} \frac{|\beta_{jk}^*| k^2}{k^2} \leq B \sum_{j \in S} \sqrt{\left(\sum_{k=d_n+1}^{\infty} \beta_{jk}^{*2} k^4 \right)} \sqrt{\left(\sum_{k=d_n+1}^{\infty} \frac{1}{k^4} \right)} \\ &\leq sBC \sqrt{\left(\sum_{k=d_n+1}^{\infty} \frac{1}{k^4} \right)} \leq \frac{sB'}{d_n^{3/2}}, \end{aligned}$$

for some constant $B' > 0$. It follows that

$$\left| \frac{1}{n} \Psi_{jk}^T V \right| \leq \left| \frac{1}{n} \sum_i \Psi_{jk}(X_{ij}) \right| \|V\|_\infty \leq \frac{Ds}{d_n^{3/2}}, \quad (72)$$

where D denotes a generic constant. Thus,

$$T_2 := \left\| \Sigma_{SS}^{-1} \left(\frac{1}{n} \Psi_S^T V \right) \right\|_\infty \leq \|\Sigma_{SS}^{-1}\|_\infty \frac{Ds}{d_n^{3/2}}. \quad (73)$$

A.2.4. Bounding T_1

Let $Z = T_1 = \Sigma_{SS}^{-1}(1/n)\Psi_S^T W$. Note that $W \sim N(0, \sigma^2 I)$, so that Z is Gaussian as well, with mean 0. Consider its l th component, $Z_l = e_l^T Z$. Then $\mathbb{E}(Z_l) = 0$, and

$$\text{var}(Z_l) = \frac{\sigma^2}{n} e_l^T \Sigma_{SS}^{-1} e_l \leq \frac{\sigma^2}{C_{\min} n}.$$

By Gaussian comparison results (Ledoux and Talagrand, 1991), we have then that

$$\mathbb{E}(\|Z\|_\infty) \leq 3\sqrt{\{\log(sd_n)\} \text{var}(Z)} \leq 3\sigma \sqrt{\left\{ \frac{\log(sd_n)}{n C_{\min}} \right\}}. \quad (74)$$

Substituting the bounds for T_2 and T_3 from equations (73) and (71) respectively into equation (69), and using the bound for the expected value of T_1 from inequality (74), it follows from an application of Markov's inequality that

$$\begin{aligned} \mathbb{P} \left(\|\hat{\beta}_S - \beta_S^*\|_\infty > \frac{\rho_n^*}{2} \right) &\leq \mathbb{P} \left\{ \|Z\|_\infty + \|\Sigma_{SS}^{-1}\|_\infty (Ds d_n^{-3/2} + \lambda_n \sqrt{C_{\max}}) > \frac{\rho_n^*}{2} \right\} \\ &\leq \frac{2}{\rho_n^*} \{ \mathbb{E}(\|Z\|_\infty) + \|\Sigma_{SS}^{-1}\|_\infty (Ds d_n^{-3/2} + \lambda_n \sqrt{C_{\max}}) \} \\ &\leq \frac{2}{\rho_n^*} \left[3\sigma \sqrt{\left\{ \frac{\log(sd_n)}{n C_{\min}} \right\}} + \|\Sigma_{SS}^{-1}\|_\infty \left(\frac{Ds}{d_n^{3/2}} + \lambda_n \sqrt{C_{\max}} \right) \right], \end{aligned}$$

which converges to 0 under the condition that

$$\frac{1}{\rho_n^*} \left[\sqrt{\left\{ \frac{\log(sd_n)}{n} \right\}} + \left\| \left(\frac{1}{n} \Psi_S^T \Psi_S \right)^{-1} \right\|_\infty \left(\frac{s}{d_n^{3/2}} + \lambda_n \right) \right] \rightarrow 0. \quad (75)$$

Noting that

$$\left\| \left(\frac{1}{n} \Psi_S^T \Psi_S \right)^{-1} \right\|_\infty \leq \frac{\sqrt{(sd_n)}}{C_{\min}}, \quad (76)$$

it follows that condition (75) holds when

$$\frac{1}{\rho_n^*} \left[\sqrt{\left\{ \frac{\log(sd_n)}{n} \right\}} + \frac{s^{3/2}}{d_n} + \lambda_n \sqrt{(sd_n)} \right] \rightarrow 0. \quad (77)$$

But this is satisfied by assumption (40) in the theorem. We have thus shown that condition (66a) is satisfied with probability converging to 1.

A.2.5. Condition (66b)

We now must consider the dual variables \hat{g}_{S^c} . Recall that we have set $\hat{\beta}_{S^c} = \beta_{S^c}^* = 0$. The stationarity condition for $j \in S^c$ is thus given by

$$\frac{1}{n} \Psi_j^T (\Psi_S \hat{\beta}_S - \Psi_S \beta_S^* - W - V) + \lambda_n \hat{g}_j = 0.$$

It then follows from equation (68) that

$$\begin{aligned}\hat{g}_{S^c} &= \frac{1}{\lambda_n} \left\{ \frac{1}{n} \Psi_{S^c}^T \Psi_S (\beta_S^* - \hat{\beta}_S) + \frac{1}{n} \Psi_{S^c}^T (W + V) \right\} \\ &= \frac{1}{\lambda_n} \left\{ \frac{1}{n} \Psi_{S^c}^T \Psi_S \left(\frac{1}{n} \Psi_S^T \Psi_S \right)^{-1} \left(\lambda_n \hat{g}_S - \frac{1}{n} \Psi_S^T W - \frac{1}{n} \Psi_S^T V \right) + \frac{1}{n} \Psi_{S^c}^T (W + V) \right\},\end{aligned}$$

so

$$\hat{g}_{S^c} = \frac{1}{\lambda_n} \left\{ \Sigma_{S^c S} \Sigma_{SS}^{-1} \left(\lambda_n \hat{g}_S - \frac{1}{n} \Psi_S^T W - \frac{1}{n} \Psi_S^T V \right) + \frac{1}{n} \Psi_{S^c}^T (W + V) \right\}. \quad (78)$$

Condition (66b) requires that

$$g_j^T \left(\frac{1}{n} \Psi_j^T \Psi_j \right)^{-1} g_j < 1, \quad (79)$$

for all $j \in S^c$. Since

$$g_j^T \left(\frac{1}{n} \Psi_j^T \Psi_j \right)^{-1} g_j \leq \frac{1}{C_{\min}} \|g_j\|^2 \quad (80)$$

it suffices to show that $\max_{j \in S^c} \|g_j\| < \sqrt{C_{\min}}$. From equation (78), we see that \hat{g}_j is Gaussian, with mean μ_j as

$$\mu_j = \mathbb{E}(\hat{g}_j) = \Sigma_{js} \Sigma_{SS}^{-1} \left(\hat{g}_S - \frac{1}{\lambda_n} \frac{1}{n} \Psi_S^T V \right) - \frac{1}{\lambda_n} \frac{1}{n} \Psi_j^T V.$$

This can be bounded as

$$\begin{aligned}\|\mu_j\| &\leq \|\Sigma_{js} \Sigma_{SS}^{-1}\| \left(\|\hat{g}_S\| + \frac{1}{\lambda_n} \left\| \frac{1}{n} \Psi_S^T V \right\| \right) + \frac{1}{\lambda_n} \left\| \frac{1}{n} \Psi_j^T V \right\| \\ &= \|\Sigma_{js} \Sigma_{SS}^{-1}\| \left\{ \sqrt{(s C_{\max})} + \frac{1}{\lambda_n} \left\| \frac{1}{n} \Psi_S^T V \right\| \right\} + \frac{1}{\lambda_n} \left\| \frac{1}{n} \Psi_j^T V \right\|.\end{aligned} \quad (81)$$

Using the bound $\|\Psi_j^T V\|_\infty \leq Ds/d_n^{3/2}$ from equation (72), we have

$$\left\| \frac{1}{n} \Psi_j^T V \right\| \leq \sqrt{d_n} \left\| \frac{1}{n} \Psi_j^T V \right\|_\infty \leq \frac{Ds}{d_n},$$

and hence

$$\left\| \frac{1}{n} \Psi_j^T V \right\| \leq \sqrt{s} \left\| \frac{1}{n} \Psi_j^T V \right\|_\infty \leq \frac{Ds^{3/2}}{d_n}.$$

Substituting in the bound (81) on the mean μ_j ,

$$\|\mu_j\| \leq \|\Sigma_{js} \Sigma_{SS}^{-1}\| \left\{ \sqrt{(s C_{\max})} + \frac{Ds^{3/2}}{\lambda_n d_n} \right\} + \frac{Ds}{\lambda_n d_n}. \quad (82)$$

Assumptions (37) and (38) of the theorem can be rewritten as

$$\|\Sigma_{js} \Sigma_{SS}^{-1}\| \leq \sqrt{\left(\frac{C_{\min}}{C_{\max}} \right) \frac{1-\delta}{\sqrt{s}}} \quad \text{for some } \delta > 0, \quad (83)$$

$$\frac{s}{\lambda_n d_n} \rightarrow 0. \quad (84)$$

Thus the bound on the mean becomes

$$\|\mu_j\| \leq \sqrt{C_{\min}(1-\delta)} + \frac{2Ds}{\lambda_n d_n} < \sqrt{C_{\min}},$$

for sufficiently large n . It therefore suffices, for condition (66b) to be satisfied, to show that

$$\mathbb{P}\left(\max_{j \in S^c} \|\hat{g}_j - \mu_j\|_\infty > \frac{\delta}{2\sqrt{d_n}}\right) \rightarrow 0, \quad (85)$$

since this implies that

$$\begin{aligned} \|\hat{g}_j\| &\leq \|\mu_j\| + \|\hat{g}_j - \mu_j\| \\ &\leq \|\mu_j\| + \sqrt{d_n} \|\hat{g}_j - \mu_j\|_\infty \\ &\leq \sqrt{C_{\min}(1-\delta)} + \frac{\delta}{2} + o(1), \end{aligned}$$

with probability approaching 1. To show result (85), we again appeal to Gaussian comparison results. Define

$$Z_j = \Psi_j^T (I - \Psi_S (\Psi_S^T \Psi_S)^{-1} \Psi_S^T) \frac{W}{n}, \quad (86)$$

for $j \in S^c$. Then Z_j are zero-mean Gaussian random variables, and we need to show that

$$\mathbb{P}\left\{\max_{j \in S^c} \left(\frac{\|Z_j\|_\infty}{\lambda_n}\right) \geq \frac{\delta}{2\sqrt{d_n}}\right\} \rightarrow \infty. \quad (87)$$

A calculation shows that $\mathbb{E}(Z_{jk}^2) \leq \sigma^2/n$. Therefore, we have by Markov's inequality and Gaussian comparison that

$$\begin{aligned} \mathbb{P}\left\{\max_{j \in S^c} \left(\frac{\|Z_j\|_\infty}{\lambda_n}\right) \geq \frac{\delta}{2\sqrt{d_n}}\right\} &\leq \frac{2\sqrt{d_n}}{\delta \lambda_n} \mathbb{E}(\max_{jk} |Z_{jk}|) \\ &\leq \frac{2\sqrt{d_n}}{\delta \lambda_n} [3\sqrt{\log\{(p-s)d_n\}} \max_{jk} \{\sqrt{\mathbb{E}(Z_{jk}^2)}\}] \\ &\leq \frac{6\sigma}{\delta \lambda_n} \sqrt{\left[\frac{d_n \log\{(p-s)d_n\}}{n}\right]}, \end{aligned}$$

which converges to 0 given the assumption (39) of the theorem that

$$\frac{\lambda_n^2 n}{d_n \log\{(p-s)d_n\}} \rightarrow \infty.$$

Thus condition (66b) is also satisfied with probability converging to 1, which completes the proof.

A.3. Proof of proposition 1

For any index k we have that

$$\|f\|_2^2 = \sum_{i=1}^{\infty} \beta_i^2 \quad (88)$$

$$\leq \|\beta\|_\infty \sum_{i=1}^{\infty} |\beta_i| \quad (89)$$

$$= \|\beta\|_\infty \sum_{i=1}^k |\beta_i| + \|\beta\|_\infty \sum_{i=k+1}^{\infty} |\beta_i| \quad (90)$$

$$\leq k \|\beta\|_\infty^2 + \|\beta\|_\infty \sum_{i=k+1}^{\infty} \frac{i^\nu |\beta_i|}{i^\nu} \quad (91)$$

$$\leq k\|\beta\|_\infty^2 + \|\beta\|_\infty \sqrt{\left(\sum_{i=1}^{\infty} \beta_i^2 i^{2\nu}\right)} \sqrt{\left(\sum_{i=k+1}^{\infty} \frac{1}{i^{2\nu}}\right)} \quad (92)$$

$$\leq k\|\beta\|_\infty^2 + \|\beta\|_\infty C \sqrt{\left(\frac{k^{1-2\nu}}{2\nu-1}\right)}, \quad (93)$$

where the last inequality uses the bound

$$\sum_{i=k+1}^{\infty} i^{-2\nu} \leq \int_k^{\infty} x^{-2\nu} dx = \frac{k^{1-2\nu}}{2\nu-1}. \quad (94)$$

Let k^* be the index that minimizes expression (93). Some calculus shows that k^* satisfies

$$c_1 \|\beta\|_\infty^{-2/(2\nu+1)} \leq k^* \leq c_2 \|\beta\|_\infty^{-2/(2\nu+1)} \quad (95)$$

for some constants c_1 and c_2 . Using the above expression in expression (93) then yields

$$\|f\|_2^2 \leq \|\beta\|_\infty (c_2 \|\beta\|_\infty^{(2\nu-1)/(2\nu+1)} + c'_1 \|\beta\|_\infty^{(2\nu-1)/(2\nu+1)}) \quad (96)$$

$$= c \|\beta\|_\infty^{4\nu/(2\nu+1)} \quad (97)$$

for some constant c , and the result follows.

A.4. Proof of theorem 3

We begin with some notation. If \mathcal{M} is a class of functions then the L_∞ bracketing number $N_{[]}(\varepsilon, \mathcal{M})$ is defined as the smallest number of pairs $B = \{(l_1, u_1), \dots, (l_k, u_k)\}$ such that $\|u_j - l_j\|_\infty \leq \varepsilon$, $1 \leq j \leq k$, and such that for every $m \in \mathcal{M}$ there exists $(l, u) \in B$ such that $l \leq m \leq u$. For the Sobolev space \mathcal{T}_j ,

$$\log\{N_{[]}(\varepsilon, \mathcal{T}_j)\} \leq K \left(\frac{1}{\varepsilon}\right)^{1/2} \quad (98)$$

for some $K > 0$; see van der Vaart (1998). The bracketing integral is defined to be

$$J_{[]}(\delta, \mathcal{M}) = \int_0^\delta \sqrt{\log\{N_{[]}(\varepsilon, \mathcal{M})\}} d\varepsilon. \quad (99)$$

From corollary 19.35 of van der Vaart (1998),

$$\mathbb{E} \left\{ \sup_{g \in \mathcal{M}} |\hat{\mu}(g) - \mu(g)| \right\} \leq \frac{C J_{[]}(\|F\|_\infty, \mathcal{M})}{\sqrt{n}} \quad (100)$$

for some $C > 0$, where $F(x) = \sup_{g \in \mathcal{M}} |g(x)|$, $\mu(g) = \mathbb{E}\{g(X)\}$ and $\hat{\mu}(g) = n^{-1} \sum_{i=1}^n g(X_i)$.

Set $Z \equiv (Z_0, \dots, Z_p) = (Y, X_1, \dots, X_p)$ and note that

$$R(\beta, g) = \sum_{j=0}^p \sum_{k=0}^p \beta_j \beta_k \mathbb{E}\{g_j(Z_j) g_k(Z_k)\} \quad (101)$$

where we define $g_0(z_0) = z_0$ and $\beta_0 = -1$. Also define

$$\hat{R}(\beta, g) = \frac{1}{n} \sum_{i=1}^n \sum_{j=0}^p \sum_{k=0}^p \beta_j \beta_k g_j(Z_{ij}) g_k(Z_{ik}). \quad (102)$$

Hence \hat{m}_n is the minimizer of $\hat{R}(\beta, g)$ subject to the constraint $\sum_j \beta_j g_j(x_j) \in \mathcal{M}_n(L_n)$ and $g_j \in \mathcal{T}_j$. For all (β, g) ,

$$|\hat{R}(\beta, g) - R(\beta, g)| \leq \|\beta\|_1^2 \max_{jk} \sup_{g_j \in \mathcal{S}_j, g_k \in \mathcal{S}_k} |\hat{\mu}_{jk}(g) - \mu_{jk}(g)| \quad (103)$$

where

$$\hat{\mu}_{jk}(g) = n^{-1} \sum_{i=1}^n \sum_{jk} g_j(Z_{ij}) g_k(Z_{ik})$$

and $\mu_{jk}(g) = \mathbb{E}\{g_j(Z_j) g_k(Z_k)\}$. From inequality (98) it follows that

$$\log\{N_{[]}(\varepsilon, \mathcal{M}_n)\} \leq 2 \log(p_n) + K \left(\frac{1}{\varepsilon}\right)^{1/2}. \quad (104)$$

Hence, $J_{[]}(\mathcal{C}, \mathcal{M}_n) = O\{\sqrt{\log(p_n)}\}$ and it follows from inequality (100) and Markov's inequality that

$$\max_{jk} \sup_{g_j \in \mathcal{S}_j, g_k \in \mathcal{S}_k} |\hat{\mu}_{jk}(g) - \mu_{jk}(g)| = O_P \left[\sqrt{\left\{ \frac{\log(p_n)}{n} \right\}} \right] = O_P \left(\frac{1}{n^{(1-\xi)/2}} \right). \quad (105)$$

We conclude that

$$\sup_{g \in \mathcal{M}} |\hat{R}(g) - R(g)| = O_P \left(\frac{L_n^2}{n^{(1-\xi)/2}} \right). \quad (106)$$

Therefore,

$$\begin{aligned} R(m^*) &\leq R(\hat{m}_n) \leq \hat{R}(\hat{m}_n) + O_P \left(\frac{L_n^2}{n^{(1-\xi)/2}} \right) \\ &\leq \hat{R}(m^*) + O_P \left(\frac{L_n^2}{n^{(1-\xi)/2}} \right) \leq R(m^*) + O_P \left(\frac{L_n^2}{n^{(1-\xi)/2}} \right) \end{aligned}$$

and the conclusion follows.

References

- Antoniadis, A. and Fan, J. (2001) Regularized wavelet approximations (with discussion). *J. Am. Statist. Ass.*, **96**, 939–967.
- Buja, A., Hastie, T. and Tibshirani, R. (1989) Linear smoothers and additive models. *Ann. Statist.*, **17**, 453–510.
- Bunea, F., Tsybakov, A. and Wegkamp, M. (2007) Sparsity oracle inequalities for the lasso. *Electron. J. Statist.*, **1**, 169–194.
- Daubechies, I., Defrise, M. and DeMol, C. (2004) An iterative thresholding algorithm for linear inverse problems. *Communs Pure Appl. Math.*, **57**, 1413–1457.
- Daubechies, I., Fornasier, M. and Loris, I. (2007) Accelerated projected gradient method for linear inverse problems with sparsity constraints. *Technical Report*. Princeton University, Princeton. (Available from arXiv:0706.4297.)
- Fan, J. and Jiang, J. (2005) Nonparametric inference for additive models. *J. Am. Statist. Ass.*, **100**, 890–907.
- Fan, J. and Li, R. Z. (2001) Variable selection via penalized likelihood. *J. Am. Statist. Ass.*, **96**, 1348–1360.
- Greenshtein, E. and Ritov, Y. (2004) Persistency in high dimensional linear predictor-selection and the virtue of over-parametrization. *Bernoulli*, **10**, 971–988.
- Hastie, T. and Tibshirani, R. (1999) *Generalized Additive Models*. New York: Chapman and Hall.
- Juditsky, A. and Nemirovski, A. (2000) Functional aggregation for nonparametric regression. *Ann. Statist.*, **28**, 681–712.
- Koltchinskii, V. and Yuan, M. (2008) Sparse recovery in large ensembles kernel machines. In *Proc. 21st A. Conf. Learning Theory*, pp. 229–238. Eastbourne: Omnipress.
- Ledoux, M. and Talagrand, M. (1991) *Probability in Banach Spaces: Isoperimetry and Processes*. New York: Springer.
- Lin, Y. and Zhang, H. H. (2006) Component selection and smoothing in multivariate nonparametric regression. *Ann. Statist.*, **34**, 2272–2297.
- Meier, L., van de Geer, S. and Bühlmann, P. (2008) High-dimensional additive modelling. (Available from arXiv.)
- Meinshausen, N. and Bühlmann, P. (2006) High dimensional graphs and variable selection with the lasso. *Ann. Statist.*, **34**, 1436–1462.
- Meinshausen, N. and Yu, B. (2006) Lasso-type recovery of sparse representations for high-dimensional data. *Technical Report 720*. Department of Statistics, University of California, Berkeley.
- Olshausen, B. A. and Field, D. J. (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, **381**, 607–609.
- Ravikumar, P., Liu, H., Lafferty, J. and Wasserman, L. (2008) Spam: sparse additive models. In *Advances in Neural Information Processing Systems*, vol. 20 (eds J. Platt, D. Koller, Y. Singer and S. Roweis), pp. 1201–1208. Cambridge: MIT Press.
- Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, **58**, 267–288.
- van der Vaart, A. W. (1998) *Asymptotic Statistics*. Cambridge: Cambridge University Press.

- Wainwright, M. (2006) Sharp thresholds for high-dimensional and noisy recovery of sparsity. *Technical Report 709*. Department of Statistics, University of California, Berkeley.
- Wainwright, M. J., Ravikumar, P. and Lafferty, J. D. (2007) High-dimensional graphical model selection using l_1 -regularized logistic regression. In *Advances in Neural Information Processing Systems*, vol. 19 (eds B. Schölkopf, J. Platt and T. Hoffman), pp. 1465–1472. Cambridge: MIT Press.
- Wasserman, L. and Roeder, K. (2007) Multi-stage variable selection: screen and clean. Carnegie Mellon University, Pittsburgh. (Available from arXiv:0704.1139.)
- Yuan, M. (2007) Nonnegative garrote component selection in functional ANOVA models. *Proc. Artif. Intell. Statist.* (Available from www.stat.umn.edu/~aistat/proceedings.)
- Yuan, M. and Lin, Y. (2006) Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, **68**, 49–67.
- Zhao, P. and Yu, B. (2007) On model selection consistency of lasso. *J. Mach. Learn. Res.*, **7**, 2541–2567.
- Zou, H. (2005) The adaptive lasso and its oracle properties. *J. Am. Statist. Ass.*, **101**, 1418–1429.

Linear Classification

36-708

In these notes we discuss parametric classification, in particular, linear classification, from several different points of view. We begin with a review of classification.

1 Review of Classification

The problem of predicting a discrete random variable Y from another random variable X is called *classification*, also sometimes called *discrimination*, *pattern classification* or *pattern recognition*. We observe iid data $(X_1, Y_1), \dots, (X_n, Y_n) \sim P$ where $X_i \in \mathbb{R}^d$ and $Y_i \in \{0, 1, \dots, K - 1\}$. Often, the covariates X are also called *features*. The goal is to predict Y given a new X ; here are some examples:

1. The Iris Flower study. The data are 50 samples from each of three species of Iris flowers, *Iris setosa*, *Iris virginica* and *Iris versicolor*; see Figure 1. The length and width of the sepal and petal are measured for each specimen, and the task is to predict the species of a new Iris flower based on these features.
2. The Coronary Risk-Factor Study (CORIS). The data consist of attributes of 462 males between the ages of 15 and 64 from three rural areas in South Africa. The outcome Y is the presence ($Y = 1$) or absence ($Y = 0$) of coronary heart disease and there are 9 covariates: systolic blood pressure, cumulative tobacco (kg), ldl (low density lipoprotein cholesterol), adiposity, famhist (family history of heart disease), typea (type-A behavior), obesity, alcohol (current alcohol consumption), and age. The goal is to predict Y from all these covariates.
3. Handwriting Digit Recognition. Here each Y is one of the ten digits from 0 to 9. There are 256 covariates X_1, \dots, X_{256} corresponding to the intensity values of the pixels in a 16×16 image; see Figure 2.
4. Political Blog Classification. A collection of 403 political blogs were collected during two months before the 2004 presidential election. The goal is to predict whether a blog is *liberal* ($Y = 0$) or *conservative* ($Y = 1$) given the content of the blog.

A *classification rule*, or *classifier*, is a function $h : \mathcal{X} \rightarrow \{0, \dots, K - 1\}$ where \mathcal{X} is the domain of X . When we observe a new X , we predict Y to be $h(X)$. Intuitively, the classification rule h partitions the input space \mathcal{X} into K disjoint *decision regions* whose boundaries are called *decision boundaries*. In these notes, we consider *linear classifiers* whose decision boundaries are linear functions of the covariate X . For $K = 2$, we have a *binary classification* problem.



Figure 1: Three different species of the Iris data. *Iris setosa* (Left), *Iris versicolor* (Middle), and *Iris virginica* (Right).

For $K > 2$, we have a *multiclass classification* problem. To simplify the discussion, we mainly discuss binary classification, and briefly explain how methods can extend to the multiclass case.

A binary classifier h is a function from \mathcal{X} to $\{0, 1\}$. It is linear if there exists a function $H(x) = \beta_0 + \beta^T x$ such that $h(x) = I(H(x) > 0)$. $H(x)$ is also called a *linear discriminant function*. The decision boundary is therefore defined as the set $\{x \in \mathbb{R}^d : H(x) = 0\}$, which corresponds to a $(d - 1)$ -dimensional hyperplane within the d -dimensional input space \mathcal{X} .

The *classification risk*, or *error rate*, of h is defined as

$$R(h) = \mathbb{P}(Y \neq h(X)) \quad (1)$$

and the *empirical classification error* or *training error* is

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^n I(h(X_i) \neq Y_i). \quad (2)$$

Here is some notation that we will use.

X	covariate (feature)
\mathcal{X}	domain of X , usually $\mathcal{X} \subset \mathbb{R}^d$
Y	response (pattern)
h	binary classifier, $h : \mathcal{X} \rightarrow \{0, 1\}$
H	linear discriminant function, $H(x) = \beta_0 + \beta^T x$ and $h(x) = I(H(x) > 0)$
m	regression function, $m(x) = \mathbb{E}(Y X = x) = \mathbb{P}(Y = 1 X = x)$
P_X	marginal distribution of X
p_j	$p_j(x) = p(x Y = j)$, the conditional density ¹ of X given that $Y = j$
π_1	$\pi_1 = \mathbb{P}(Y = 1)$
P	joint distribution of (X, Y)

Now we review some key results.

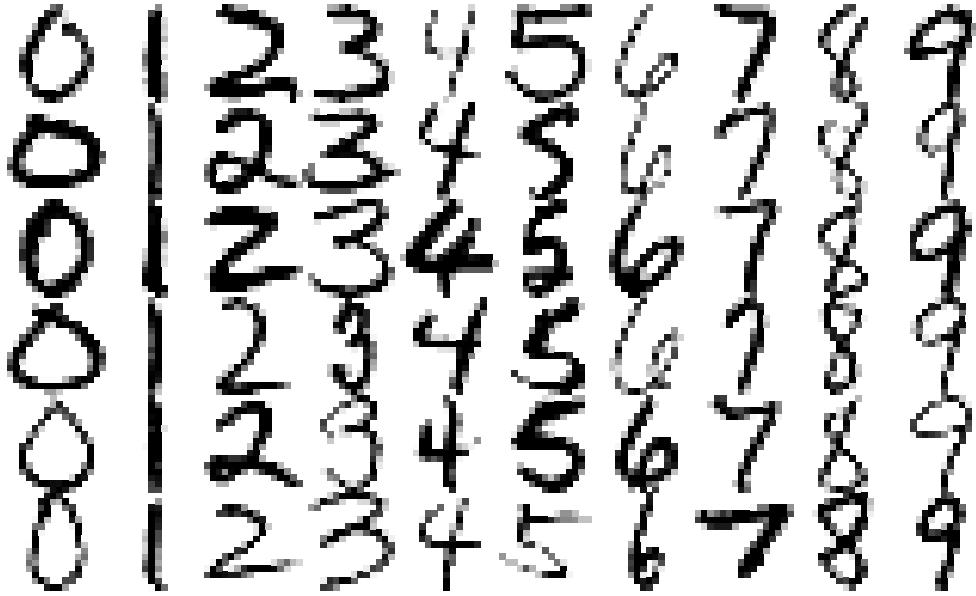


Figure 2: Examples from the zipcode data.

Theorem 1 *The rule h that minimizes $R(h)$ is*

$$h^*(x) = \begin{cases} 1 & \text{if } m(x) > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $m(x) = \mathbb{E}(Y|X = x) = \mathbb{P}(Y = 1|X = x)$ denotes the regression function.

The rule h^* is called the *Bayes rule*. The risk $R^* = R(h^*)$ of the Bayes rule is called the *Bayes risk*. The set $\{x \in \mathcal{X} : m(x) = 1/2\}$ is called the *Bayes decision boundary*.

Proof. We will show that $R(h) - R(h^*) \geq 0$. Note that

$$R(h) = \mathbb{P}(\{Y \neq h(X)\}) = \int \mathbb{P}(Y \neq h(X)|X = x)dP_X(x).$$

It suffices to show that

$$\mathbb{P}(Y \neq h(X)|X = x) - \mathbb{P}(Y \neq h^*(X)|X = x) \geq 0 \quad \text{for all } x \in \mathcal{X}. \quad (4)$$

Now,

$$\mathbb{P}(Y \neq h(X)|X = x) = 1 - \mathbb{P}(Y = h(X)|X = x) \quad (5)$$

$$= 1 - (\mathbb{P}(Y = 1, h(X) = 1|X = x) + \mathbb{P}(Y = 0, h(X) = 0|X = x)) \quad (6)$$

$$= 1 - (h(x)\mathbb{P}(Y = 1|X = x) + (1 - h(x))\mathbb{P}(Y = 0|X = x)) \quad (7)$$

$$= 1 - (h(x)m(x) + (1 - h(x))(1 - m(x))). \quad (8)$$

Hence,

$$\begin{aligned}
& \mathbb{P}(Y \neq h(X)|X = x) - \mathbb{P}(Y \neq h^*(X)|X = x) \\
&= \left(h^*(x)m(x) + (1 - h^*(x))(1 - m(x)) \right) - \left(h(x)m(x) + (1 - h(x))(1 - m(x)) \right) \\
&= (2m(x) - 1)(h^*(x) - h(x)) = 2 \left(m(x) - \frac{1}{2} \right) (h^*(x) - h(x)). \tag{9}
\end{aligned}$$

When $m(x) \geq 1/2$ and $h^*(x) = 1$, (9) is non-negative. When $m(x) < 1/2$ and $h^*(x) = 0$, (9) is again non-negative. This proves (4). \square

We can rewrite h^* in a different way. From Bayes' theorem

$$\begin{aligned}
m(x) &= \mathbb{P}(Y = 1|X = x) = \frac{p(x|Y = 1)\mathbb{P}(Y = 1)}{p(x|Y = 1)\mathbb{P}(Y = 1) + p(x|Y = 0)\mathbb{P}(Y = 0)} \\
&= \frac{\pi_1 p_1(x)}{\pi_1 p_1(x) + (1 - \pi_1)p_0(x)}. \tag{10}
\end{aligned}$$

where $\pi_1 = \mathbb{P}(Y = 1)$. From the above equality, we have that

$$m(x) > \frac{1}{2} \text{ is equivalent to } \frac{p_1(x)}{p_0(x)} > \frac{1 - \pi_1}{\pi_1}. \tag{11}$$

Thus the Bayes rule can be rewritten as

$$h^*(x) = \begin{cases} 1 & \text{if } \frac{p_1(x)}{p_0(x)} > \frac{1 - \pi_1}{\pi_1} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

If \mathcal{H} is a set of classifiers then the classifier $h_o \in \mathcal{H}$ that minimizes $R(h)$ is the *oracle classifier*. Formally,

$$R(h_o) = \inf_{h \in \mathcal{H}} R(h)$$

and $R_o = R(h_o)$ is called the *oracle risk* of \mathcal{H} . In general, if h is any classifier and R^* is the Bayes risk then,

$$R(h) - R^* = \underbrace{R(h) - R(h_o)}_{\text{distance from oracle}} + \underbrace{R(h_o) - R^*}_{\text{distance of oracle from Bayes error}}. \tag{13}$$

The first term is analogous to the variance, and the second is analogous to the squared bias in linear regression.

For a binary classifier problem, given a covariate X we only need to predict its class label $Y = 0$ or $Y = 1$. This is in contrast to a regression problem where we need to predict a real-valued response $Y \in \mathbb{R}$. Intuitively, classification is a much easier task than regression. To rigorously formalize this, let $m^*(x) = \mathbb{E}(Y|X = x)$ be the true regression function and

let $h^*(x)$ be the corresponding Bayes rule. Let $\hat{m}(x)$ be an estimate of $m^*(x)$ and define the *plug-in classification rule*:

$$\hat{h}(x) = \begin{cases} 1 & \text{if } \hat{m}(x) > \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

We have the following theorem.

Theorem 2 *The risk of the plug-in classifier rule in (14) satisfies*

$$R(\hat{h}) - R^* \leq 2 \sqrt{\int (\hat{m}(x) - m^*(x))^2 dP_X(x)}.$$

Proof. In the proof of Theorem 1 we showed that

$$\begin{aligned} \mathbb{P}(Y \neq \hat{h}(X)|X=x) - \mathbb{P}(Y \neq h^*(X)|X=x) &= (2\hat{m}(x) - 1)(h^*(x) - \hat{h}(x)) \\ &= |2\hat{m}(x) - 1|I(h^*(x) \neq \hat{h}(x)) = 2|\hat{m}(x) - 1/2|I(h^*(x) \neq \hat{h}(x)). \end{aligned}$$

Now, when $h^*(x) \neq \hat{h}(x)$, there are two possible cases: (i) $\hat{h}(x) = 1$ and $h^*(x) = 0$; (ii) $\hat{h}(x) = 0$ and $h^*(x) = 1$. In both cases, we have that $|\hat{m}(x) - m^*(x)| \geq |\hat{m}(x) - 1/2|$. Therefore,

$$\begin{aligned} \mathbb{P}(\hat{h}(X) \neq Y) - \mathbb{P}(h^*(X) \neq Y) &= 2 \int |\hat{m}(x) - 1/2|I(h^*(x) \neq \hat{h}(x))dP_X(x) \\ &\leq 2 \int |\hat{m}(x) - m^*(x)|I(h^*(x) \neq \hat{h}(x))dP_X(x) \\ &\leq 2 \int |\hat{m}(x) - m^*(x)|dP_X(x) \end{aligned} \quad (15)$$

$$\leq 2 \sqrt{\int (\hat{m}(x) - m^*(x))^2 dP_X(x)}. \quad (16)$$

The last inequality follows from the fact that $\mathbb{E}|Z| \leq \sqrt{\mathbb{E}Z^2}$ for any Z . \square

This theorem implies that if the regression estimate $\hat{m}(x)$ is close to $m^*(x)$ then the plug-in classification risk will be close to the Bayes risk. The converse is *not* necessarily true. It is possible for \hat{m} to be far from $m^*(x)$ and still lead to a good classifier. As long as $\hat{m}(x)$ and $m^*(x)$ are on the same side of 1/2 they yield the same classifier.

Example 3 *Figure 3 shows two one-dimensional regression functions. In both cases, the Bayes rule is $h^*(x) = I(x > 0)$ and the decision boundary is $\mathcal{D} = \{x = 0\}$. The left plot illustrates an easy problem; there is little ambiguity around the decision boundary. Even a poor estimate of $m(x)$ will recover the correct decision boundary. The right plot illustrates a hard problem; it is hard to know from the data if you are to the left or right of the decision boundary.*

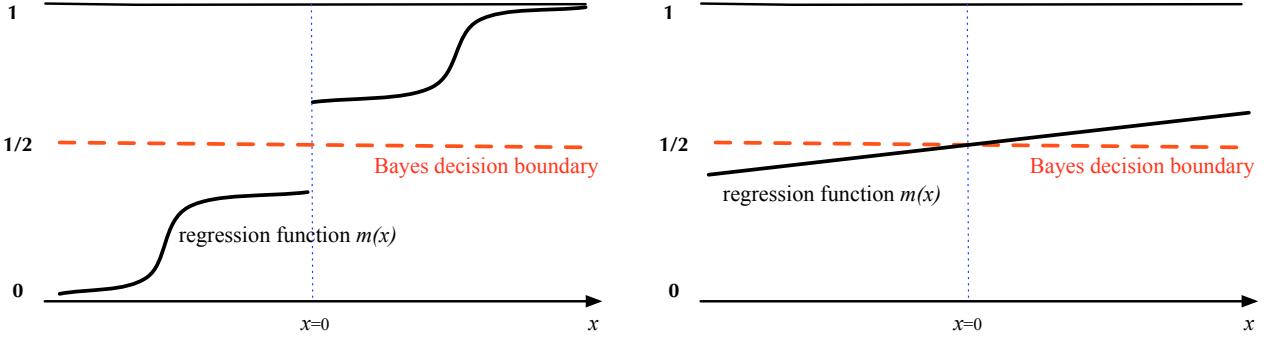


Figure 3: The Bayes rule is $h^*(x) = I(x > 0)$ in both plots, which show the regression function $m(x) = \mathbb{E}(Y|x)$ for two problems. The left plot shows an easy problem; there is little ambiguity around the decision boundary. The right plot shows a hard problem; it is hard to know from the data if you are to the left or right of the decision boundary.

So classification is easier than regression. Can it be strictly easier? Suppose that $R(\hat{m}) \rightarrow 0$. We have that

$$\begin{aligned}
R(\hat{h}) - R(h_*) &\leq 2 \int |\hat{m}(x) - m_*(x)| I(h_*(x) \neq \hat{h}(x)) dP(x) \\
&= 2 \int |\hat{m}(x) - m_*(x)| I(h_*(x) \neq \hat{h}(x)) I(m_*(x) \neq 1/2) dP(x) \\
&= 2\mathbb{E} \left[|\hat{m}(X) - m_*(X)| I(h_*(X) \neq \hat{h}(X)) I(m_*(X) \neq 1/2) \right] \\
&\leq 2\mathbb{E} \left[|\hat{m}(X) - m_*(X)| I(h_*(X) \neq \hat{h}(X)) I(|m_*(X) - 1/2| \leq \epsilon, m_*(X) \neq 1/2) \right] \\
&\quad + 2\mathbb{E} \left[|\hat{m}(X) - m_*(X)| I(h_*(X) \neq \hat{h}(X)) I(|m_*(X) - 1/2| > \epsilon) \right] \\
&\leq 2\sqrt{R(\hat{m})}(a^{1/2} + b^{1/2})
\end{aligned}$$

where

$$a = P(h_*(X) \neq \hat{h}(X), |m_*(X) - 1/2| \leq \epsilon, m_*(X) \neq 1/2)$$

and

$$b = P(h_*(X) \neq \hat{h}(X), |m_*(X) - 1/2| > \epsilon).$$

Now

$$b \leq P(|\hat{m}(X) - m_*(X)| > \epsilon) \leq \frac{R(\hat{m})}{\epsilon^2} \rightarrow 0$$

so

$$\lim_{n \rightarrow \infty} \frac{R(\hat{h}) - R(h_*)}{\sqrt{R(\hat{m})}} \leq 2a^{1/2}.$$

But $a \rightarrow 0$ as $\epsilon \rightarrow 0$ so So

$$\frac{R(\hat{h}) - R(h_*)}{\sqrt{R(\hat{m})}} \rightarrow 0.$$

So the LHS can be smaller than the right hand side. But how much smaller? Yang (1999) showed that if the class of regression functions is sufficiently rich, then

$$\inf_{\hat{m}} \sup_{m \in \mathcal{M}} R(\hat{h}) \asymp r_n^2 \quad \text{and} \quad \inf_{\hat{h}} \sup_{m \in \mathcal{M}} [R(\hat{h}) - R(h_*)] \asymp r_n$$

which says that the minimax classification rate is the square root of the regression rate.

But, there are natural classes that fail the richness condition such as low noise classes. For example, if $P(|m_*(X) - 1/2| \leq \epsilon) = 0$ and \hat{m} satisfies an exponential inequality then $\frac{R(\hat{h}) - R(h_*)}{\sqrt{R(\hat{m})}}$ is exponentially small. So it really depends on the problem.

2 Empirical Risk Minimization

The conceptually simplest approach is empirical risk minimization (ERM) where we minimize the training error over all linear classifiers. Let $H_\beta(x) = \beta^T x$ (where $x(1) = 1$) and $h_\beta(x) = I(H_\beta(x) > 0)$. Thus we define $\hat{\beta}$ to be the value of β that minimizes

$$\hat{R}_n(\beta) = \frac{1}{n} \sum_{i=1}^n I(Y_i \neq h_\beta(X_i)).$$

The problem with this approach is that it is difficult to minimize $\hat{R}_n(\beta)$. The theory for the ERM is straightforward. First, let us recall the following. Let \mathcal{H} be a set of classifiers and let $h_* = \operatorname{argmin}_{h \in \mathcal{H}} R(h)$. Let \hat{h} minimize the empirical risk. If $\sup_{h \in \mathcal{H}} |\hat{R}(h) - R(h)| \leq \epsilon$ then $R(\hat{h}) \leq R(h_*) + 2\epsilon$. To see this, note that if $\sup_{h \in \mathcal{H}} |\hat{R}(h) - R(h)| \leq \epsilon$ then, using the fact that \hat{h} minimizes \hat{R} ,

$$R(h_*) \leq R(\hat{h}) \leq \hat{R}(\hat{h}) + \epsilon \leq \hat{R}(h_*) + \epsilon \leq R(h_*) + 2\epsilon.$$

So we need to bound

$$P(\sup_{h \in \mathcal{H}} |\hat{R}(h) - R(h)| \leq \epsilon).$$

If \mathcal{H} has finite VC dimension r then

$$P(\sup_{h \in \mathcal{H}} |\hat{R}(h) - R(h)| \leq \epsilon) \leq 8(n+1)^r e^{-n\epsilon^2/32}.$$

Now half-spaces have VC dimension $r = d+1$. So

$$P(\sup_{h \in \mathcal{H}} |\hat{R}(h) - R(h)| \leq \epsilon) \leq 8(n+1)^{d+1} e^{-n\epsilon^2/32}.$$

We conclude that $P(R(\hat{h}) - R(h_*) > 2\epsilon) \leq 8n^{d+1} e^{-n\epsilon^2/32}$.

The result can be improved if there are not too many data points near the decision boundary. We'll state a result due to Koltchinski and Panchenko (2002) that involves *the margin*. (See also Kakade, Sridharan and Tewari 2009). Let us take $Y_i \in \{-1, +1\}$ so we can write $h(x) = \text{sign}(\beta^T x)$. Suppose that $|X(j)| \leq A < \infty$ for each j . We also restrict ourselves to the set of linear classifiers $h(x) = \text{sign}(\beta^T x)$ with $|\beta(j)| \leq A$. Define the *margin-sensitive loss*

$$\phi_\gamma(u) = \begin{cases} 1 & \text{if } u \leq 0 \\ 1 - \frac{u}{\gamma} & \text{if } 0 < u \leq \gamma \\ 0 & \text{if } u > \gamma. \end{cases}$$

Then, for any such classifier h , with probability at least $1 - \delta$,

$$P(Y \neq h(X)) \leq \frac{1}{n} \sum_{i=1}^n \phi_\gamma(Y_i h(X_i)) + \frac{4A^{3/2}d}{\gamma n} + \left(\frac{8}{\gamma} + 1 \right) \sqrt{\frac{\log(4/\delta)}{2n}}.$$

This means that, if there are few observations near the boundary, then, by taking γ large, we can make the loss small. However, the restriction to bounded covariates and bounded classifiers is non-trivial.

3 Gaussian Discriminant Analysis

Suppose that $p_0(x) = p(x|Y = 0)$ and $p_1(x) = p(x|Y = 1)$ are both multivariate Gaussians:

$$p_k(x) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right\}, \quad k = 0, 1.$$

where Σ_1 and Σ_2 are both $d \times d$ covariance matrices. Thus, $X|Y = 0 \sim N(\mu_0, \Sigma_0)$ and $X|Y = 1 \sim N(\mu_1, \Sigma_1)$.

Given a square matrix A , we define $|A|$ to be the determinant of A . For a binary classification problem with Gaussian distributions, we have the following theorem.

Theorem 4 *If $X|Y = 0 \sim N(\mu_0, \Sigma_0)$ and $X|Y = 1 \sim N(\mu_1, \Sigma_1)$, then the Bayes rule is*

$$h^*(x) = \begin{cases} 1 & \text{if } r_1^2 < r_0^2 + 2 \log \left(\frac{\pi_1}{1-\pi_1} \right) + \log \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

where $r_i^2 = (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)$ for $i = 1, 2$ is the Mahalanobis distance.

Proof. By definition, the Bayes rule is $h^*(x) = I(\pi_1 p_1(x) > (1 - \pi_1) p_0(x))$. Plug-in the specific forms of p_0 and p_1 and take the logarithms we get $h^*(x) = 1$ if and only if

$$\begin{aligned} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) - 2 \log \pi_1 + \log(|\Sigma_1|) \\ < (x - \mu_0)^T \Sigma_0^{-1} (x - \mu_0) - 2 \log(1 - \pi_1) + \log(|\Sigma_0|). \end{aligned} \quad (18)$$

The theorem immediately follows from some simple algebra. \square

Let $\pi_0 = 1 - \pi_1$. An equivalent way of expressing the Bayes rule is

$$h^*(x) = \operatorname{argmax}_{k \in \{0,1\}} \delta_k(x) \quad (19)$$

where

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log \pi_k \quad (20)$$

is called the *Gaussian discriminant function*. The decision boundary of the above classifier can be characterized by the set $\{x \in \mathcal{X} : \delta_1(x) = \delta_0(x)\}$, which is quadratic so this procedure is called *quadratic discriminant analysis* (QDA).

In practice, we use sample quantities of $\pi_0, \pi_1, \mu_1, \mu_2, \Sigma_0, \Sigma_1$ in place of their population values, namely

$$\hat{\pi}_0 = \frac{1}{n} \sum_{i=1}^n (1 - Y_i), \quad \hat{\pi}_1 = \frac{1}{n} \sum_{i=1}^n Y_i, \quad (21)$$

$$\hat{\mu}_0 = \frac{1}{n_0} \sum_{i: Y_i=0} X_i, \quad \hat{\mu}_1 = \frac{1}{n_1} \sum_{i: Y_i=1} X_i, \quad (22)$$

$$\hat{\Sigma}_0 = \frac{1}{n_0 - 1} \sum_{i: Y_i=0} (X_i - \hat{\mu}_0)(X_i - \hat{\mu}_0)^T, \quad (23)$$

$$\hat{\Sigma}_1 = \frac{1}{n_1 - 1} \sum_{i: Y_i=1} (X_i - \hat{\mu}_1)(X_i - \hat{\mu}_1)^T, \quad (24)$$

where $n_0 = \sum_i (1 - Y_i)$ and $n_1 = \sum_i Y_i$. (Note: we could also estimate Σ_0 and Σ_1 using their maximum likelihood estimates, which replace $n_0 - 1$ and $n_1 - 1$ with n_0 and n_1 .)

A simplification occurs if we assume that $\Sigma_0 = \Sigma_1 = \Sigma$. In this case, the Bayes rule is

$$h^*(x) = \operatorname{argmax}_k \delta_k(x) \quad (25)$$

where now

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k. \quad (26)$$

Hence, the classifier is linear. The parameters are estimated as before, except that we use a pooled estimate of the Σ :

$$\hat{\Sigma} = \frac{(n_0 - 1)\hat{\Sigma}_0 + (n_1 - 1)\hat{\Sigma}_1}{n_0 + n_1 - 2}. \quad (27)$$

The classification rule is

$$h^*(x) = \begin{cases} 1 & \text{if } \delta_1(x) > \delta_0(x) \\ 0 & \text{otherwise.} \end{cases} \quad (28)$$

The decision boundary $\{x \in \mathcal{X} : \delta_0(x) = \delta_1(x)\}$ is linear so this method is called *linear discrimination analysis* (LDA).

When the dimension d is large, fully specifying the QDA decision boundary requires $d + d(d - 1)$ parameters, and fully specifying the LDA decision boundary requires $d + d(d - 1)/2$ parameters. Such a large number of free parameters might induce a large variance. To further regularize the model, two popular methods are *diagonal quadratic discriminant analysis* (DQDA) and *diagonal linear discriminant analysis* (DLDA). The only difference between DQDA and DLDA with QDA and LDA is that after calculating $\widehat{\Sigma}_1$ and $\widehat{\Sigma}_0$ as in (24), we set all the off-diagonal elements to be zero. This is also called the independence rule.

We now generalize to the case where Y takes on more than two values. That is, $Y \in \{0, \dots, K - 1\}$ for $K > 2$. First, we characterize the Bayes classifier under this multiclass setting.

Theorem 5 Let $R(h) = \mathbb{P}(h(X) \neq Y)$ be the classification error of a classification rule $h(x)$. The Bayes rule $h^*(X)$ minimizing $R(h)$ can be written as

$$h^*(x) = \operatorname{argmax}_k \mathbb{P}(Y = k | X = x) \quad (29)$$

Proof. We have

$$R(h) = 1 - \mathbb{P}(h(X) = Y) \quad (30)$$

$$= 1 - \sum_{k=0}^{K-1} \mathbb{P}(h(X) = k, Y = k) \quad (31)$$

$$= 1 - \sum_{k=0}^{K-1} \mathbb{E}\left[I(h(X) = k) \mathbb{P}(Y = k | X)\right] \quad (32)$$

It's clear that $h^*(X) = \operatorname{argmax}_k \mathbb{P}(Y = k | X)$ achieves the minimized classification error $1 - \mathbb{E}[\max_k \mathbb{P}(Y = k | X)]$. \square

Let $\pi_k = \mathbb{P}(Y = k)$. The next theorem extends QDA and LDA to the multiclass setting.

Theorem 6 Suppose that $Y \in \{0, \dots, K - 1\}$ with $K \geq 2$. If $p_k(x) = p(x | Y = k)$ is Gaussian : $X | Y = k \sim N(\mu_k, \Sigma_k)$, the Bayes rule for the multiclass QDA can be written as

$$h^*(x) = \operatorname{argmax}_k \delta_k(x)$$

where

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log \pi_k. \quad (33)$$

If all Gaussians have an equal variance Σ , then

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k. \quad (34)$$

Let $n_k = \sum_i I(y_i = k)$ for $k = 0, \dots, K - 1$. The estimated sample quantities of π_k , μ_k , Σ_k , and Σ are:

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n I(y_i = k), \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{i: Y_i=k} X_i, \quad (35)$$

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i: Y_i=k} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^T, \quad (36)$$

$$\hat{\Sigma} = \frac{\sum_{k=0}^{K-1} (n_k - 1) \hat{\Sigma}_k}{n - K}. \quad (37)$$

Example 7 Let us return to the Iris data example. Recall that there are 150 observations made on three classes of the iris flower: Iris setosa, Iris versicolor, and Iris virginica. There are four features: sepal length, sepal width, petal length, and petal width. In Figure 4 we visualize the datasets. Within each class, we plot the densities for each feature. It's easy to see that the distributions of petal length and petal width are quite different across different classes, which suggests that they are very informative features.

Figures 5 and 6 provide multiple figure arrays illustrating the classification of observations based on LDA and QDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to these a given pair of features before fitting the model. We see that the decision boundaries for LDA are linear, while the decision boundaries for QDA are highly nonlinear. The training errors for LDA and QDA on this data are both 0.02. From these figures, we see that it is very easy to discriminate the observations of class Iris setosa from those of the other two classes.

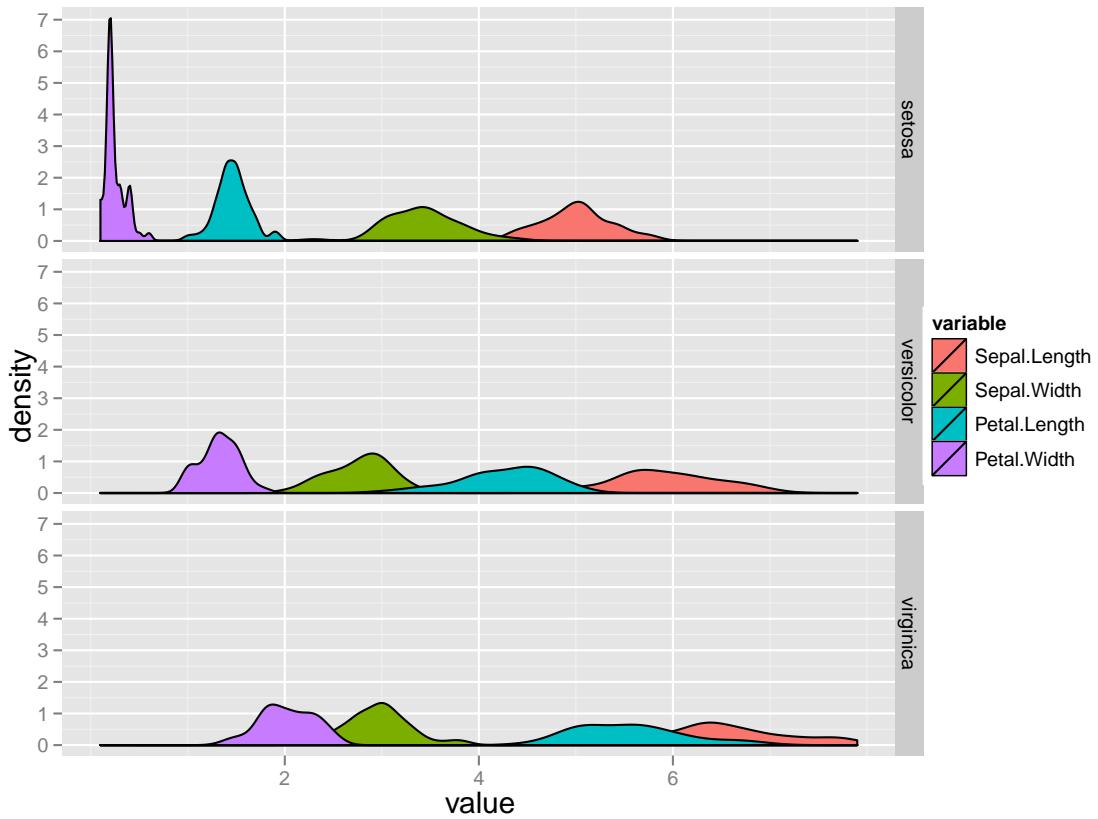


Figure 4: The Iris data: The estimated densities for different features are plotted within each class. It's easy to see that the distributions of petal length and petal width are quite different across different classes, which suggests that they are very informative features.

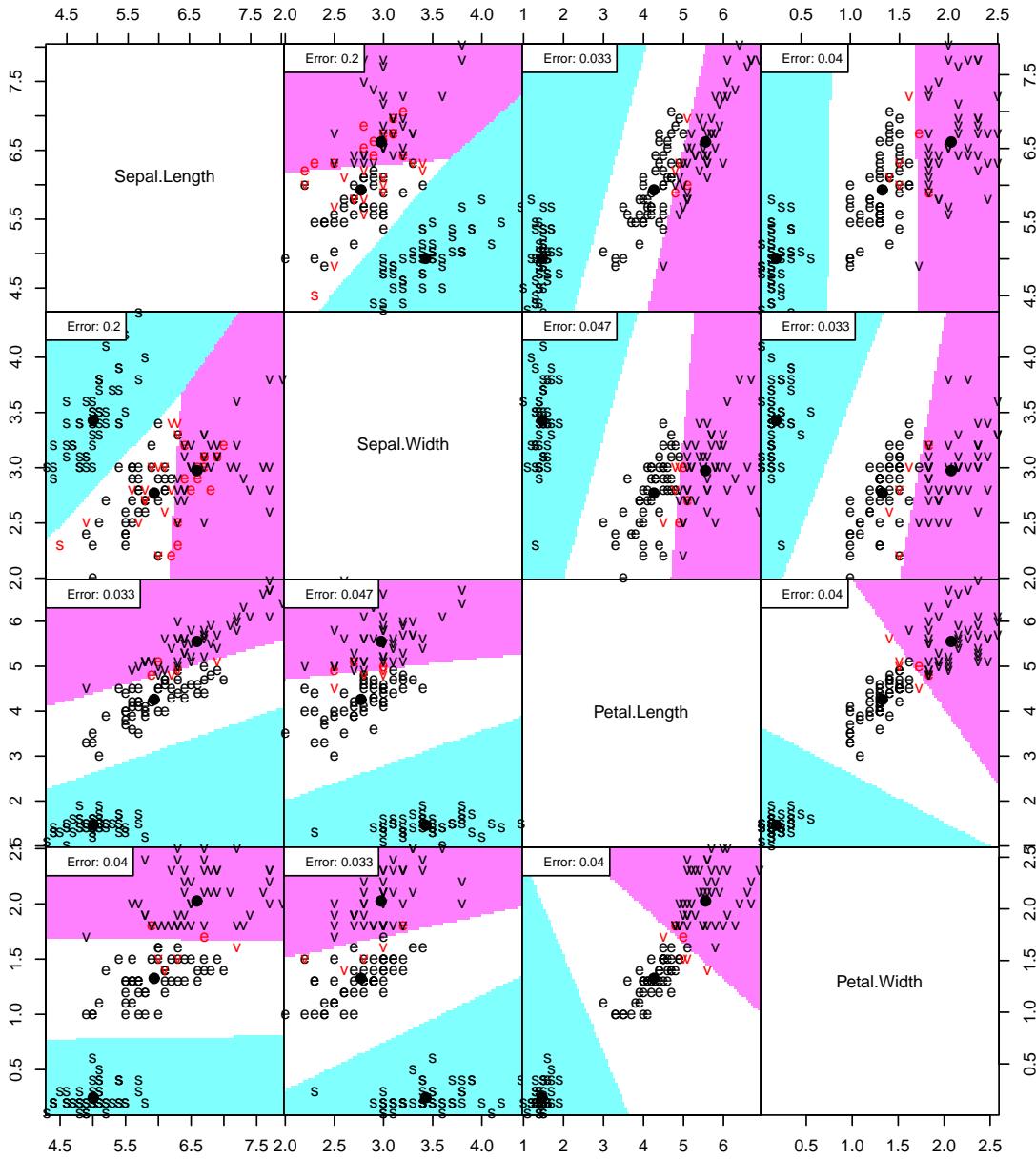


Figure 5: Classifying the Iris data using LDA. The multiple figure array illustrates the classification of observations based on LDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to a given pair of features before fitting the model. In these plots, “s” represents the class label *Iris setosa*, “e” represents the class label *Iris versicolor*, and “v” represents the class label *Iris virginica*. The red letters illustrate the misclassified observations.

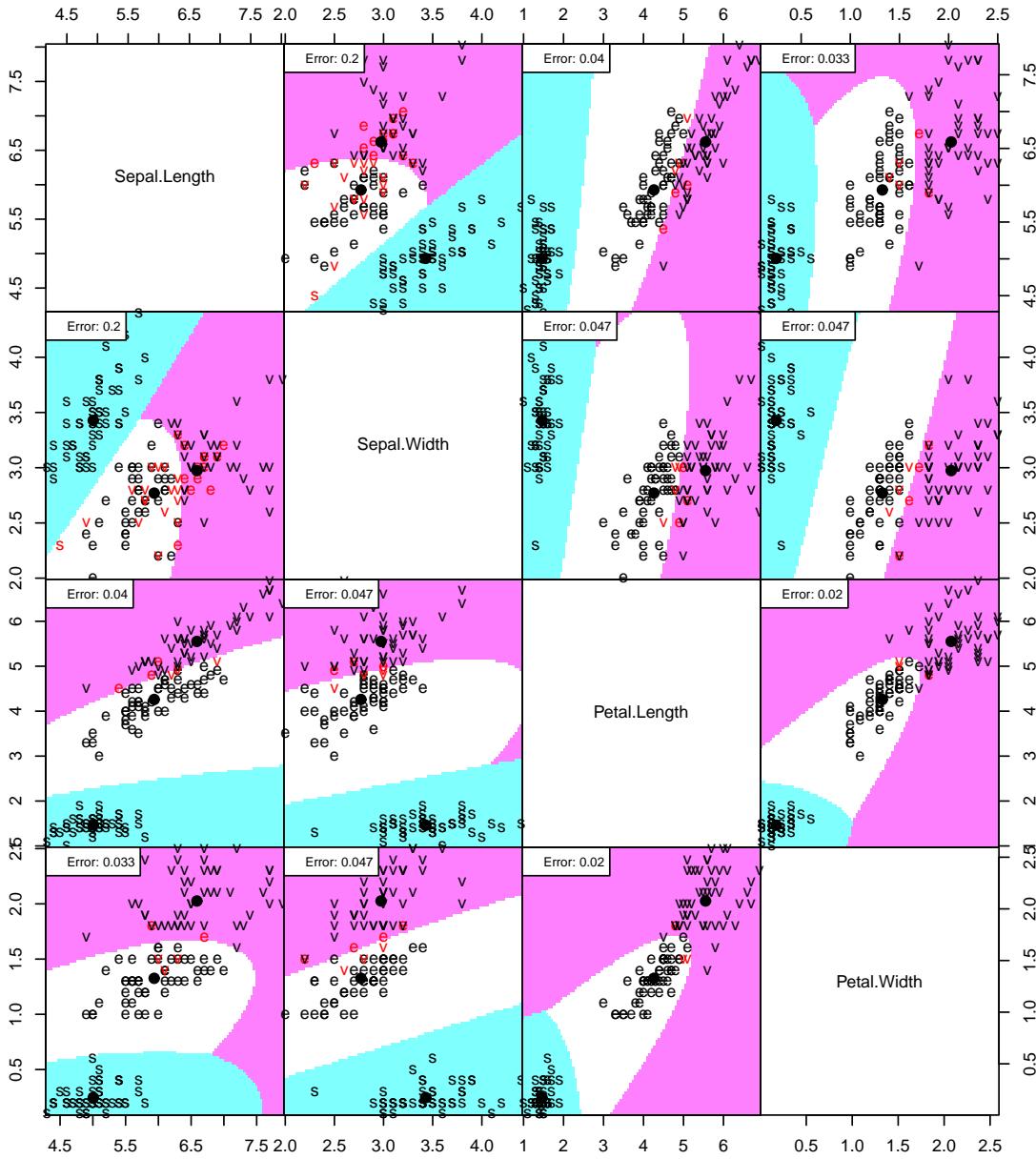


Figure 6: Classifying the Iris data using QDA. The multiple figure array illustrates the classification of observations based on QDA for every combination of two features. The classification boundaries are displayed and the classification error by simply casting the data onto these two features are calculated. In these plots, “s” represents the class label *Iris setosa*, “e” represents the class label *Iris versicolor*, and “v” represents the class label *Iris virginica*. The red letters illustrate the misclassified observations.

4 Fisher Linear Discriminant Analysis

There is another version of linear discriminant analysis due to Fisher (1936). The idea is to first reduce the covariates to one dimension by projecting the data onto a line. Algebraically, this means replacing the covariate $X = (X_1, \dots, X_d)^T$ with a linear combination $U = w^T X = \sum_{j=1}^d w_j X_j$. The goal is to choose the vector $w = (w_1, \dots, w_d)^T$ that “best separates the data into two groups.” Then we perform classification with the one-dimensional covariate U instead of X .

What do we mean by “best separates the data into two groups”? Formally, we would like the two groups to have means that as far apart as possible relative to their spread. Let μ_j denote the mean of X for $Y = j$, $j = 0, 1$. And let Σ be the covariance matrix of X . Then, for $j = 0, 1$, $\mathbb{E}(U|Y = j) = \mathbb{E}(w^T X|Y = j) = w^T \mu_j$ and $\text{Var}(U) = w^T \Sigma w$. Define the separation by

$$\begin{aligned} J(w) &= \frac{(\mathbb{E}(U|Y = 0) - \mathbb{E}(U|Y = 1))^2}{w^T \Sigma w} \\ &= \frac{(w^T \mu_0 - w^T \mu_1)^2}{w^T \Sigma w} \\ &= \frac{w^T (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T \Sigma w}. \end{aligned}$$

J is sometimes called the Rayleigh coefficient. Our goal is to find w that maximizes $J(w)$. Since $J(w)$ involves unknown population quantities $\Sigma_0, \Sigma_1, \mu_0, \mu_1$, we estimate J as follows. Let $n_j = \sum_{i=1}^n I(Y_i = j)$ be the number of observations in class j , let $\hat{\mu}_j$ be the sample mean vector of the X ’s for class j , and let Σ_j be the sample covariance matrix for all observations in class j . Define

$$\hat{J}(w) = \frac{w^T S_B w}{w^T S_W w} \tag{38}$$

where

$$\begin{aligned} S_B &= (\hat{\mu}_0 - \hat{\mu}_1)(\hat{\mu}_0 - \hat{\mu}_1)^T, \\ S_W &= \frac{(n_0 - 1)S_0 + (n_1 - 1)S_1}{(n_0 - 1) + (n_1 - 1)}. \end{aligned}$$

Theorem 8 *The vector*

$$\hat{w} = S_W^{-1}(\hat{\mu}_0 - \hat{\mu}_1) \tag{39}$$

is a maximizer of $\hat{J}(w)$.

Proof. Maximizing $\hat{J}(w)$ is equivalent to maximizing $w^T S_B w$ subject to the constraint that $w^T S_W w = 1$. This is a generalized eigenvalue problem. By the definition of eigenvector and

eigenvalue, the maximizer \hat{w} should be the eigenvector of $S_W^{-1}S_B$ corresponding to the largest eigenvalue. The key observation is that $S_B = (\hat{\mu}_0 - \hat{\mu}_1)(\hat{\mu}_0 - \hat{\mu}_1)^T$, which implies that for any vector w , $S_B w$ must be in the direction of $\hat{\mu}_0 - \hat{\mu}_1$. The desired result immediately follows. \square

We call

$$f(x) = \hat{w}^T x = (\hat{\mu}_0 - \hat{\mu}_1)^T S_W^{-1} x \quad (40)$$

the *Fisher linear discriminant function*. Given a cutting threshold $c_m \in \mathbb{R}$, Fisher's classification rule is

$$h(x) = \begin{cases} 0 & \text{if } \hat{w}^T x \geq c_m \\ 1 & \text{if } \hat{w}^T x < c_m. \end{cases} \quad (41)$$

Fisher's rule is the same as the Gaussian LDA rule in (26) when

$$c_m = \frac{1}{2}(\hat{\mu}_0 - \hat{\mu}_1)^T S_W^{-1}(\hat{\mu}_0 + \hat{\mu}_1) - \log\left(\frac{\hat{\pi}_0}{\hat{\pi}_1}\right). \quad (42)$$

5 Logistic Regression

One approach to binary classification is to estimate the regression function $m(x) = \mathbb{E}(Y|X=x) = \mathbb{P}(Y=1|X=x)$ and, once we have an estimate $\hat{m}(x)$, use the classification rule

$$\hat{h}(x) = \begin{cases} 1 & \text{if } \hat{m}(x) > \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \quad (43)$$

For binary classification problems, one possible choice is the linear regression model

$$Y = m(X) + \epsilon = \beta_0 + \sum_{j=1}^d \beta_j X_j + \epsilon. \quad (44)$$

The linear regression model does not explicitly constrain Y to take on binary values. A more natural alternative is to use *logistic regression*, which is the most common binary classification method.

Before we describe the logistic regression model, let's recall some basic facts about binary random variables. If Y takes values 0 and 1, we say that Y has a Bernoulli distribution with parameter $\pi_1 = \mathbb{P}(Y=1)$. The probability mass function for Y is $p(y; \pi_1) = \pi_1^y(1-\pi_1)^{1-y}$ for $y = 0, 1$. The likelihood function for π_1 based on iid data Y_1, \dots, Y_n is

$$\mathcal{L}(\pi_1) = \prod_{i=1}^n p(Y_i; \pi_1) = \prod_{i=1}^n \pi_1^{Y_i}(1-\pi_1)^{1-Y_i}. \quad (45)$$

In the logistic regression model, we assume that

$$m(x) = \mathbb{P}(Y = 1|X = x) = \frac{\exp(\beta_0 + x^T \beta)}{1 + \exp(\beta_0 + x^T \beta)} \equiv \pi_1(x, \beta_0, \beta). \quad (46)$$

In other words, given $X = x$, Y is Bernoulli with mean $\pi_1(x, \beta_0, \beta)$. We can write the model as

$$\text{logit}(\mathbb{P}(Y = 1|X = x)) = \beta_0 + x^T \beta \quad (47)$$

where $\text{logit}(a) = \log(a/(1 - a))$. The name ‘‘logistic regression’’ comes from the fact that $\exp(x)/(1 + \exp(x))$ is called the logistic function.

Lemma 9 *Both linear regression and logistic regression models have linear decision boundaries.*

Proof. The linear decision boundary for linear regression is straightforward. The same result for logistic regression follows from the monotonicity of the logistic function. \square

The parameters β_0 and $\beta = (\beta_1, \dots, \beta_d)^T$ can be estimated by maximum conditional likelihood. The conditional likelihood function for β is

$$\mathcal{L}(\beta_0, \beta) = \prod_{i=1}^n \pi(x_i, \beta_0, \beta)^{Y_i} (1 - \pi(x_i, \beta_0, \beta))^{1-Y_i}.$$

Thus the conditional log-likelihood is

$$\ell(\beta_0, \beta) = \sum_{i=1}^n \left\{ Y_i \log \pi(x_i, \beta_0, \beta) - (1 - y_i) \log(1 - \pi(x_i, \beta_0, \beta)) \right\} \quad (48)$$

$$= \sum_{i=1}^n \left\{ Y_i(\beta_0 + x_i^T \beta) - \log(1 + \exp(\beta_0 + x_i^T \beta)) \right\}. \quad (49)$$

The maximum conditional likelihood estimators $\hat{\beta}_0$ and $\hat{\beta}$ cannot be found in closed form. However, the loglikelihood function is concave and can be efficiently solve by the Newton’s method in an iterative manner as follows.

Note that the logistic regression classifier is essentially replacing the 0-1 loss with a smooth loss function. In other words, it uses a *surrogate loss function*.

For notational simplicity, we redefine (local to this section) the d -dimensional covariate x_i and parameter vector β as the following $(d + 1)$ -dimensional vectors:

$$x_i \leftarrow (1, x_i^T)^T \text{ and } \beta \leftarrow (\beta_0, \beta^T)^T. \quad (50)$$

Thus, we write $\pi_1(x, \beta_0, \beta)$ as $\pi_1(x, \beta)$ and $\ell(\beta_0, \beta)$ as $\ell(\beta)$.

To maximize $\ell(\beta)$, the $(k+1)$ th Newton step in the algorithm replaces the k th iterate $\widehat{\beta}^{(k)}$ by

$$\widehat{\beta}^{(k+1)} \leftarrow \widehat{\beta}^{(k)} - \left(\frac{\partial^2 \ell(\widehat{\beta}^{(k)})}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\widehat{\beta}^{(k)})}{\partial \beta}. \quad (51)$$

The gradient $\partial s \frac{\partial \ell(\widehat{\beta}^{(k)})}{\partial \beta}$ and Hessian $\partial s \frac{\partial^2 \ell(\widehat{\beta}^{(k)})}{\partial \beta \partial \beta^T}$ are both evaluated at $\widehat{\beta}^{(k)}$ and can be written as

$$\frac{\partial \ell(\widehat{\beta}^{(k)})}{\partial \beta} = \sum_{i=1}^n (\pi(x_i, \widehat{\beta}^{(k)}) - Y_i) X_i \text{ and } \frac{\partial^2 \ell(\widehat{\beta}^{(k)})}{\partial \beta \partial \beta^T} = -\mathbb{X}^T \mathbb{W} \mathbb{X} \quad (52)$$

where $\mathbb{W} = \text{diag}(w_{11}^{(k)}, w_{22}^{(k)}, \dots, w_{dd}^{(k)})$ is a diagonal matrix with

$$w_{ii}^{(k)} = \pi(x_i, \widehat{\beta}^{(k)}) (1 - \pi(x_i, \widehat{\beta}^{(k)})). \quad (53)$$

Let $\pi_1^{(k)} = (\pi_1(x_1, \widehat{\beta}^{(k)}), \dots, \pi_1(x_n, \widehat{\beta}^{(k)}))^T$, (51) can be written as

$$\widehat{\beta}^{(k+1)} = \widehat{\beta}^{(k)} + (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T (y - \pi_1^{(k)}) \quad (54)$$

$$= (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T \mathbb{W} (\mathbb{X} \widehat{\beta}^{(k)} + \mathbb{W}^{-1} (y - \pi_1^{(k)})) \quad (55)$$

$$= (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T \mathbb{W} z^{(k)} \quad (56)$$

where $z^{(k)} \equiv (z_1^{(k)}, \dots, z_n^{(k)})^T = \mathbb{X}^T \widehat{\beta}^{(k)} + \mathbb{W}^{-1} (y - \pi_1^{(k)})$ with

$$z_i^{(k)} = \log \left(\frac{\pi_1(x_i, \widehat{\beta}^{(k)})}{1 - \pi_1(x_i, \widehat{\beta}^{(k)})} + \frac{y_i - \pi_1(x_i, \widehat{\beta}^{(k)})}{\pi_1(x_i, \widehat{\beta}^{(k)}) (1 - \pi_1(x_i, \widehat{\beta}^{(k)}))} \right). \quad (57)$$

Given the current estimate $\widehat{\beta}^{(k)}$, the above Newton iteration forms a quadratic approximation to the negative log-likelihood using Taylor expansion at $\widehat{\beta}^{(k)}$:

$$-\ell(\beta) = \underbrace{\frac{1}{2} (z - \mathbb{X} \beta)^T \mathbb{W} (z - \mathbb{X} \beta)}_{\ell_Q(\beta)} + \text{constant}. \quad (58)$$

The update equation (56) corresponds to solving a quadratic optimization

$$\widehat{\beta}^{(k+1)} = \underset{\beta}{\operatorname{argmin}} \ell_Q(\beta). \quad (59)$$

We then get an iterative algorithm called *iteratively reweighted least squares*. See Figure 7.

Iteratively Reweighted Least Squares Algorithm

Choose starting values $\hat{\beta}^{(0)} = (\hat{\beta}_0^{(0)}, \hat{\beta}_1^{(0)}, \dots, \hat{\beta}_d^{(0)})^T$ and compute $\pi_1(x_i, \hat{\beta}^{(0)})$ using Equation (46), for $i = 1, \dots, n$ with β_j replaced by its initial value $\hat{\beta}_j^{(0)}$.

For $k = 1, 2, \dots$, iterate the following steps until convergence.

1. Calculate $z_i^{(k)}$ according to (57) for $i = 1, \dots, n$.
2. Calculate $\hat{\beta}^{(k+1)}$ according to (56). This corresponds to doing a weighted linear regression of z on \mathbb{X} .
3. Update the $\pi(x_i, \hat{\beta})$'s using (46) with the current estimate of $\hat{\beta}^{(k+1)}$.

Figure 7: Finding the Logistic Regression MLE.

We can get the estimated standard errors of the final solution $\hat{\beta}$. For the k th iteration, recall that the Fisher information matrix $I(\hat{\beta}^{(k)})$ takes the form

$$I(\hat{\beta}^{(k)}) = -\mathbb{E} \left(\frac{\partial^2 \ell(\hat{\beta}^{(k)})}{\partial \beta \partial \beta^T} \right) \approx \mathbb{X}^T \mathbb{W} \mathbb{X}, \quad (60)$$

we estimate the standard error of $\hat{\beta}_j$ as the j th diagonal element of $I(\hat{\beta})^{-1}$.

Example 10 We apply the logistic regression on the Coronary Risk-Factor Study (CORIS) data and yields the following estimates and Wald statistics W_j for the coefficients:

Covariate	$\hat{\beta}_j$	se	W_j	p-value
Intercept	-6.145	1.300	-4.738	0.000
sbp	0.007	0.006	1.138	0.255
tobacco	0.079	0.027	2.991	0.003
ldl	0.174	0.059	2.925	0.003
adiposity	0.019	0.029	0.637	0.524
famhist	0.925	0.227	4.078	0.000
typea	0.040	0.012	3.233	0.001
obesity	-0.063	0.044	-1.427	0.153
alcohol	0.000	0.004	0.027	0.979
age	0.045	0.012	3.754	0.000

6 Logistic Regression Versus LDA

There is a close connection between logistic regression and Gaussian LDA. Let (X, Y) be a pair of random variables where Y is binary and let $p_0(x) = p(x|Y=0)$, $p_1(x) = p(x|Y=1)$, $\pi_1 = \mathbb{P}(Y=1)$. By Bayes' theorem,

$$\mathbb{P}(Y=1|X=x) = \frac{p(x|Y=1)\pi_1}{p(x|Y=1)\pi_1 + p(x|Y=0)(1-\pi_1)} \quad (61)$$

If we assume that each group is Gaussian with the same covariance matrix Σ , i.e., $X|Y=0 \sim N(\mu_0, \Sigma)$ and $X|Y=1 \sim N(\mu_1, \Sigma)$, we have

$$\log \left(\frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=0|X=x)} \right) = \log \left(\frac{\pi}{1-\pi} \right) - \frac{1}{2}(\mu_0 + \mu_1)^T \Sigma^{-1} (\mu_1 - \mu_0) \quad (62)$$

$$+ x^T \Sigma^{-1} (\mu_1 - \mu_0) \quad (63)$$

$$\equiv \alpha_0 + \alpha^T x. \quad (64)$$

On the other hand, the logistic regression model is, by assumption,

$$\log \left(\frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=0|X=x)} \right) = \beta_0 + \beta^T x.$$

These are the same model since they both lead to classification rules that are linear in x . The difference is in how we estimate the parameters.

This is an example of a generative versus a discriminative model. In Gaussian LDA we estimate the whole joint distribution by maximizing the full likelihood

$$\prod_{i=1}^n p(X_i, Y_i) = \underbrace{\prod_{i=1}^n p(X_i|Y_i)}_{\text{Gaussian}} \underbrace{\prod_{i=1}^n p(Y_i)}_{\text{Bernoulli}}. \quad (65)$$

In logistic regression we maximize the conditional likelihood $\prod_{i=1}^n p(Y_i|X_i)$ but ignore the second term $p(X_i)$:

$$\prod_{i=1}^n p(X_i, Y_i) = \underbrace{\prod_{i=1}^n p(Y_i|X_i)}_{\text{logistic}} \underbrace{\prod_{i=1}^n p(X_i)}_{\text{ignored}}. \quad (66)$$

Since classification only requires the knowledge of $p(y|x)$, we don't really need to estimate the whole joint distribution. Logistic regression leaves the marginal distribution $p(x)$ unspecified so it relies on less parametric assumption than LDA. This is an advantage of the logistic regression approach over LDA. However, if the true class conditional distributions are Gaussian, the logistic regression will be asymptotically less efficient than LDA, i.e. to achieve a certain level of classification error, the logistic regression requires more samples.

7 Regularized Logistic Regression

As with linear regression, when the dimension d of the covariate is large, we cannot simply fit a logistic model to all the variables without experiencing numerical and statistical problems. Akin to the lasso, we will use *regularized logistic regression*, which includes *sparse logistic regression* and *ridge logistic regression*.

Let $\ell(\beta_0, \beta)$ be the log-likelihood defined in (49). The *sparse logistic regression* estimator is an ℓ_1 -regularized conditional log-likelihood estimator

$$\widehat{\beta}_0, \widehat{\beta} = \operatorname{argmin}_{\beta_0, \beta} \left\{ -\ell(\beta_0, \beta) + \lambda \|\beta\|_1 \right\}. \quad (67)$$

Similarly, the *ridge logistic regression* estimator is an ℓ_2 -regularized conditional log-likelihood estimator

$$\widehat{\beta}_0, \widehat{\beta} = \operatorname{argmin}_{\beta_0, \beta} \left\{ -\ell(\beta_0, \beta) + \lambda \|\beta\|_2^2 \right\}. \quad (68)$$

The algorithm for logistic ridge regression only requires a simple modification of the iteratively reweighted least squares algorithm and is left as an exercise.

For sparse logistic regression, an easy way to calculate $\widehat{\beta}_0$ and $\widehat{\beta}$ is to apply a ℓ_1 -regularized Newton procedure. Similar to the Newton method for unregularized logistic regression, for the k th iteration, we first form a quadratic approximation to the negative log-likelihood $\ell(\beta_0, \beta)$ based on the current estimates $\widehat{\beta}^{(k)}$.

$$-\ell(\beta_0, \beta) = \underbrace{\frac{1}{2} \sum_{i=1}^n w_{ii} (z_i^{(k)} - \beta_0 - \sum_{j=1}^d \beta_j x_{ij})^2}_{\ell_Q(\beta_0, \beta)} + \text{constant.} \quad (69)$$

where w_{ii} and $z_i^{(k)}$ are defined in (53) and (57). Since we have a ℓ_1 -regularization term, the updating formula for the estimate in the $(k+1)$ th step then becomes

$$\widehat{\beta}^{(k+1)}, \widehat{\beta}^{(k+1)} = \operatorname{argmin}_{\beta_0, \beta} \left\{ \frac{1}{2} \sum_{i=1}^n w_{ii} (z_i^{(k)} - \beta_0 - \sum_{j=1}^d \beta_j x_{ij})^2 + \lambda \|\beta\|_1 \right\}. \quad (70)$$

This is a weighted lasso problem and can be solved using coordinate descent. See Figure 8.

Even though the above iterative procedure does not guarantee theoretical convergence, it works very well in practice.

Sparse Logistic Regression Using Coordinate Descent

Choose starting values $\widehat{\beta}^{(0)} = (\widehat{\beta}_0^{(0)}, \widehat{\beta}_1^{(0)}, \dots, \widehat{\beta}_d^{(0)})^T$

(Outer loop) For $k = 1, 2, \dots$, iterate the following steps until convergence.

1. For $i = 1, \dots, n$, calculate $\pi_1(x_i, \widehat{\beta}^{(k)})$, $z_i^{(k)}$, $w_{ii}^{(k)}$ according to (46), (57), and (53).
2. $\alpha_0 = \partial_S \frac{\sum_{i=1}^n w_{ii}^{(k)} z_i^{(k)}}{\sum_{i=1}^n w_{ii}^{(k)}}$ and $\alpha_\ell = \widehat{\beta}_\ell^{(k)}$ for $\ell = 1, \dots, d$.
3. (Inner loop) iterate the following steps until convergence

For $j \in \{1, \dots, d\}$

- (a) For $i = 1, \dots, n$, calculate $r_{ij} = z_i^{(k)} - \alpha_0 - \sum_{\ell \neq j} \alpha_\ell x_{i\ell}$.
- (b) Calculate $u_j^{(k)} = \sum_{i=1}^n w_{ii}^{(k)} r_{ij} x_{ij}$ and $v_j^{(k)} = \sum_{i=1}^n w_{ii}^{(k)} x_{ij}^2$.
- (c) $\alpha_j = \text{sign}(u_j^{(k)}) \left[\frac{|u_j^{(k)}| - \lambda}{v_j^{(k)}} \right]_+$.
4. $\widehat{\beta}_0^{(k+1)} = \alpha_0$ and $\widehat{\beta}_\ell^{(k+1)} = \alpha_\ell$ for $\ell = 1, \dots, d$.
5. Update the $\pi(x_i, \widehat{\beta})$'s using (46) with the current estimate of $\widehat{\beta}^{(k+1)}$.

Figure 8: Sparse Logistic Regression

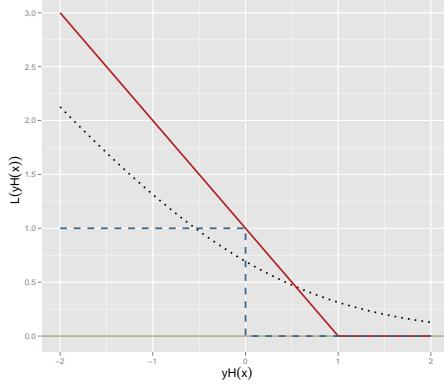


Figure 9: The 0-1 classification loss (blue dashed line), hinge loss (red solid line) and logistic loss (black dotted line).

8 Support Vector Machines

The *support vector machine (SVM)* classifier is a linear classifier that replaces the 0-1 loss with a surrogate loss function. (Logistic regression uses a different surrogate.) In this section, the outcomes are coded as -1 and $+1$. The 0-1 loss is $L(x, y, \beta) = I(y \neq h_\beta(x)) = I(yH_\beta(x) < 0)$ with the *hinge loss* $L_{\text{hinge}}(y_i, H(x_i)) \equiv [1 - Y_i H(X_i)]_+$ instead of the logistic loss. This is the smallest convex function that lies above the 0-1 loss. (When we discuss nonparameteric classifiers, we will consider more general support vector machines.)

The support vector machine classifier is $\hat{h}(x) = I(\hat{H}(x) > 0)$ where the hyperplane $\hat{H}(x) = \hat{\beta}_0 + \hat{\beta}^T x$ is obtained by minimizing

$$\sum_{i=1}^n [1 - Y_i H(X_i)]_+ + \frac{\lambda}{2} \|\beta\|_2^2 \quad (71)$$

where $\lambda > 0$ and the factor $1/2$ is only for notational convenience.

Figure 9 compares the hinge loss, 0-1 loss, and logistic loss. The advantage of the hinge loss is that it is convex, and it has a corner which leads to efficient computation and the minimizer of $\mathbb{E}(1 - Y H(X))_+$ is the Bayes rule. A disadvantage of the hinge loss is that one can't recover the regression function $m(x) = \mathbb{E}(Y|X = x)$.

The SVM classifier is often developed from a geometric perspective. Suppose first that the data are *linearly separable*, that is, there exists a hyperplane that perfectly separates the two classes. How can we find a separating hyperplane? LDA is not guaranteed to find it. A separating hyperplane will minimize

$$-\sum_{i \in \mathcal{M}} Y_i H(X_i).$$

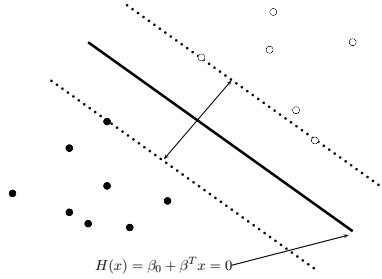


Figure 10: The hyperplane $H(x)$ has the largest margin of all hyperplanes that separate the two classes.

where \mathcal{M} is the index set of all misclassified data points. Rosenblatt's perceptron algorithm takes starting values and iteratively updates the coefficients as:

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} Y_i X_i \\ Y_i \end{pmatrix}$$

where $\rho > 0$ is the learning rate. If the data are linearly separable, the perceptron algorithm is guaranteed to converge to a separating hyperplane. However, there could be many separating hyperplanes. Different starting values may lead to different separating hyperplanes. The question is, which separating hyperplane is the best?

Intuitively, it seems reasonable to choose the hyperplane "furthest" from the data in the sense that it separates the +1's and -1's and maximizes the distance to the closest point. This hyperplane is called the *maximum margin hyperplane*. The margin is the distance from the hyperplane to the nearest data point. Points on the boundary of the margin are called *support vectors*. See Figure 10. The goal, then, is to find a separating hyperplane which maximizes the margin. After some simple algebra, we can show that (71) exactly achieves this goal. In fact, (71) also works for data that are not linearly separable.

The unconstrained optimization problem (71) can be equivalently formulated in constrained form:

$$\min_{\beta_0, \beta} \quad \left\{ \frac{1}{2} \|\beta\|_2^2 + \frac{1}{\lambda} \sum_{i=1}^n \xi_i \right\} \quad (72)$$

$$\text{subject to} \quad \forall i, \xi_i \geq 0 \text{ and } \xi_i \geq 1 - y_i H(x_i). \quad (73)$$

Given two vectors a and b , let $\langle a, b \rangle = a^T b = \sum_j a_j b_j$ denote the inner product of a and b . The following lemma provides the dual of the optimization problem in (72).

Lemma 11 *The dual of the SVM optimization problem in (72) takes the form*

$$\hat{\alpha} = \operatorname{argmax}_{\alpha \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^n \alpha_i \alpha_k Y_i y_k \langle X_i, x_k \rangle \right\} \quad (74)$$

$$\text{subject to } 0 \leq \alpha_1, \dots, \alpha_n \leq \frac{1}{\lambda} \text{ and } \sum_{i=1}^n \alpha_i y_i = 0, \quad (75)$$

with the primal-dual relationship $\hat{\beta} = \sum_{i=1}^n \hat{\alpha}_i y_i x_i$. We also have

$$\hat{\alpha}_i (1 - \xi_i - y_i (\hat{\beta}_0 + \hat{\beta}^T x_i)) = 0, \quad i = 1, \dots, n. \quad (76)$$

Proof. Let $\alpha_i, \gamma_i \geq 0$ be the Lagrange multipliers. The Lagrangian function can be written as

$$L(\xi, \beta, \beta_0, \alpha, \gamma) = \frac{1}{2} \|\beta\|_2^2 + \frac{1}{\lambda} \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - \xi_i - y_i H(x_i)) - \sum_{i=1}^n \gamma_i \xi_i. \quad (77)$$

The Karush-Kuhn-Tucker conditions are

$$\forall i, \alpha_i \geq 0, \gamma_i \geq 0, \xi_i \geq 0 \text{ and } \xi_i \geq 1 - y_i H(x_i), \quad (78)$$

$$\beta = \sum_{i=1}^n \alpha_i y_i x_i, \quad \sum_{i=1}^n \alpha_i y_i = 0 \text{ and } \alpha_i + \gamma_i = 1/\lambda, \quad (79)$$

$$\forall i, \alpha_i (1 - \xi_i - y_i H(x_i)) = 0 \text{ and } \gamma_i \xi_i = 0. \quad (80)$$

The dual formulation in (74) follows by plugging (78) and (79) into (77). The primal-dual complementary slackness condition (76) is obtained from the first equation in (80). \square

The dual problem (74) is easier to solve than the primal problem (71). The data points (X_i, y_i) for which $\hat{\alpha}_i > 0$ are called *support vectors*. By (76) and (72), for all the data points (x_i, y_i) satisfying $y_i (\hat{\beta}_0 + \hat{\beta}^T x_i) > 1$, there must be $\hat{\alpha}_i = 0$. The solution for the dual problem is sparse. From the first equality in (79), we see that the final estimate $\hat{\beta}$ is a linear combination only of these support vectors. Among these support vectors, if $\alpha_i < 1/\lambda$, we call (x_i, y_i) a *margin point*. For a margin point (x_i, y_i) , the last equality in (79) implies that $\gamma_i > 0$, then the second equality in (80) implies $\xi_i = 0$. Moreover, using the first equality in (80), we get

$$\hat{\beta}_0 = -Y_i X_i^T \hat{\beta}. \quad (81)$$

Therefore, once $\hat{\beta}$ is given, we could calculate $\hat{\beta}_0$ using any margin point (X_i, y_i) .

Example 12 *We consider classifying two types of irises, versicolor and virginica. There are 50 observations in each class. The covariates are "Sepal.Length" "Sepal.Width" "Petal.Length" and "Petal.Width". After fitting a SVM we get a 3/100 misclassification rate. The SVM uses 33 support vectors.*

9 Case Study I: Supernova Classification

A *supernova* is an exploding star. Type Ia supernovae are a special class of supernovae that are very useful in astrophysics research. These supernovae have a characteristic *light curve*, which is a plot of the luminosity of the supernova versus time. The maximum brightness of all type Ia supernovae is approximately the same. In other words, the true (or absolute) brightness of a type Ia supernova is known. On the other hand, the apparent (or observed) brightness of a supernova can be measured directly. Since we know both the absolute and apparent brightness of a type Ia supernova, we can compute its distance. Because of this, type Ia supernovae are sometimes called standard candles. Two supernovae, one type Ia and one non-type Ia, are illustrated in Figure 11. Astronomers also measure the *redshift* of the supernova, which is essentially the speed at which the supernova is moving away from us. The relationship between distance and redshift provides important information for astrophysicists in studying the large scale structure of the universe.

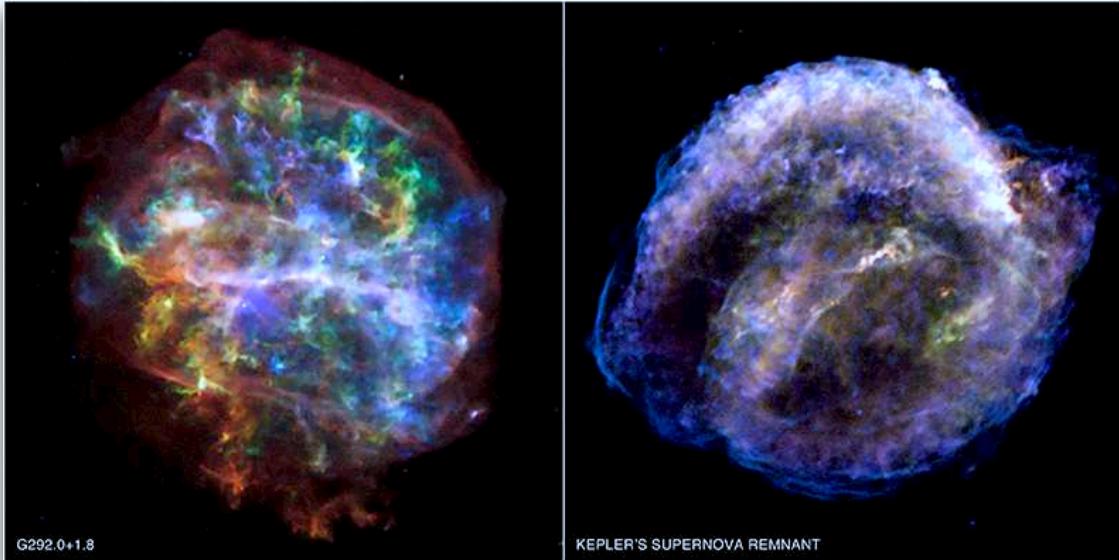


Figure 11: Two supernova remnants from the NASA’s Chandra X-ray Observatory study. The image in the right panel, the so-called Kepler supernova remnant, is ”Type Ia”. Such supernovae have a very symmetric, circular remnant. This type of supernova is thought to be caused by a thermonuclear explosion of a white dwarf, and is often used by astronomers as a “standard candle” for measuring cosmic distances. The image in the left panel is a different type of supernova that comes from “core collapse.” Such supernovae are distinctly more asymmetric. (Credit: NASA/CXC/UCSC/L. Lopez et al.)

A challenge in astrophysics is to classify supernovae to be type Ia versus other types. [?] released a mixture of real and realistically simulated supernovae and challenged the scientific community to find effective ways to classify the type Ia supernovae. The dataset consists of

about 20,000 simulated supernovae. For each supernova, there are a few noisy measurements of the flux (brightness) in four different filters. These four filters correspond to different wavelengths. Specifically, the filters correspond to the g -band (green), r -band (red), i -band (infrared) and z -band (blue). See Figure 12.

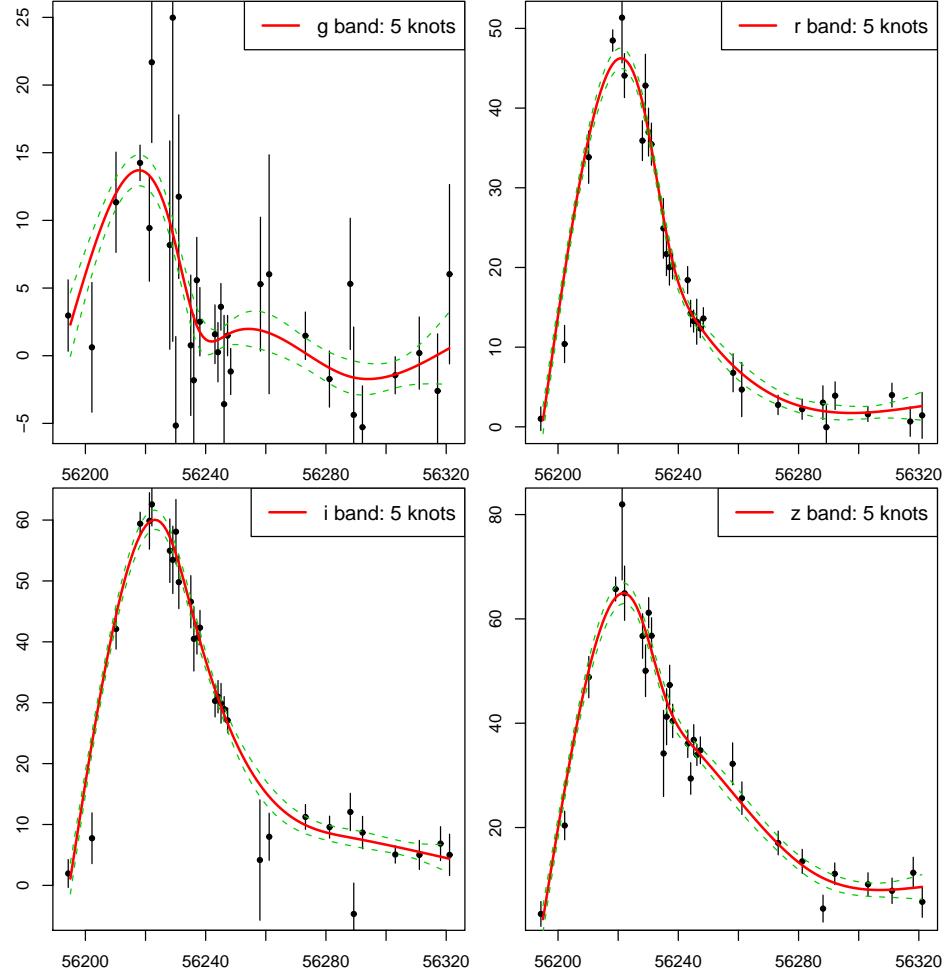


Figure 12: Four filters (g, r, i, z -bands) corresponding to a type Ia supernova *DES-SN000051*. For each band, a weighted regression spline fit (solid red) with the corresponding standard error curves (dashed green) is provided. The black points with bars represent the flux values and their estimated standard errors.

To estimate a linear classifier we need to preprocess the data to extract features. One difficulty is that each supernova is only measured at a few irregular time points, and these time points are not aligned. To handle this problem we use nonparametric regression to get a smooth curve. (We used the estimated measurement errors of each flux as weights and fitted a weighted least squares regression spline to smooth each supernova.) All four filters

of each supernova are then aligned according to the peak of the r -band. We also rescale so that all the curves have the same maximum.

The goal of this study is to build linear classifiers to predict whether a supernova is type Ia or not. For simplicity, we only use the information in the r -band. First, we align the fitted regression spline curves of all supernovae by calibrating their maximum peaks and set the corresponding time point to be day 0. There are altogether 19,679 supernovae in the dataset with 1,367 being labeled. To get a higher signal-to-noise ratio, we throw away all supernovae with less than 10 r -band flux measurements. We finally get a trimmed dataset with 255 supernovae, 206 of which are type Ia and 49 of which are non-type Ia.

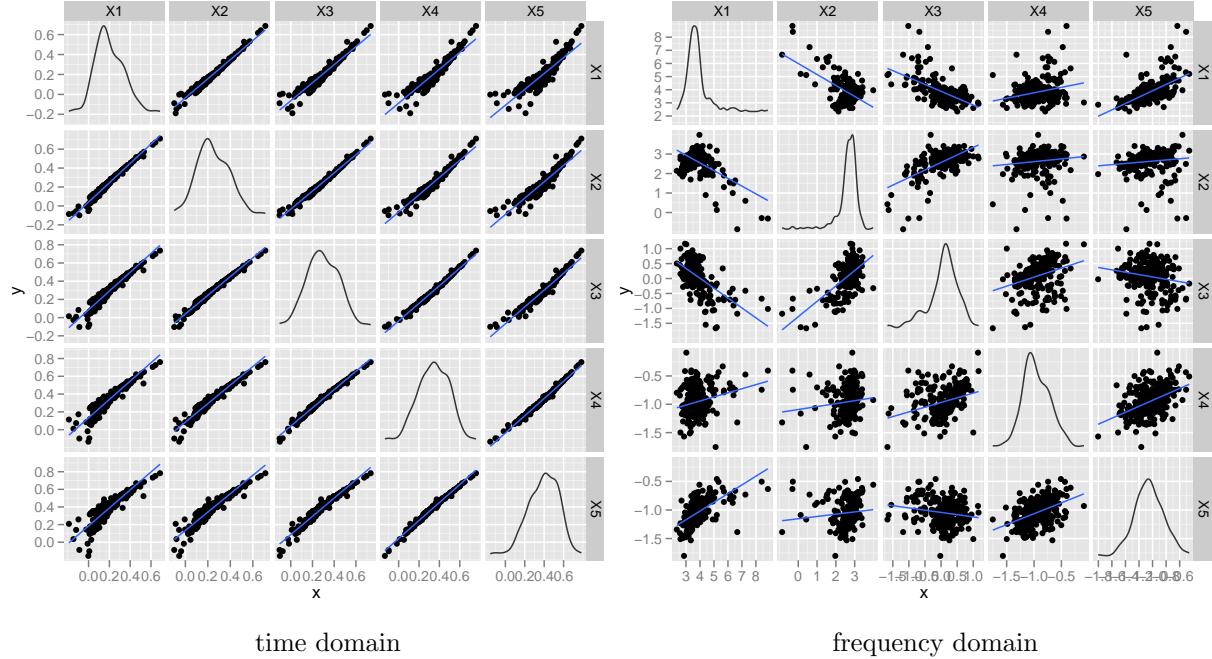


Figure 13: The matrix of scatterplots of the first five features of the supernova data. On the diagonal cells are the estimated univariate densities of each feature. The off-diagonal cells visualize the pairwise scatter plots of the two corresponding variables with a least squares fit. We see the time-domain features are highly correlated, while the frequency-domain features are almost uncorrelated.

We use two types of features: the *time-domain* features and *frequency-domain* features. For the time-domain features, the features are the interpolated regression spline values according to an equally spaced time grid. In this study, the grid has length 100, ranging from day -20 to day 80. Since all the fitted regression curves have similar global shapes, the time-domain features are expected to be highly correlated. This conjecture is confirmed by the matrix of scatterplots of the first five features in 13. To make the features less correlated, we also extract the frequency-domain features, which are simply the discrete cosine transformations of the corresponding time-domain features. More specifically, given the time domain features X_1, \dots, X_d ($d = 100$), Their corresponding frequency domain features $\tilde{X}_1, \dots, \tilde{X}_d$ can be

written as

$$\tilde{X}_j = \frac{2}{d} \sum_{k=1}^d X_k \cos\left[\frac{\pi}{d}\left(k - \frac{1}{2}\right)(j-1)\right] \text{ for } j = 1, \dots, d. \quad (82)$$

The right panel of Figure 13 illustrates the scatter matrix of the first 5 frequency-domain features. In contrast to the time-domain features, the frequency-domain features have low correlation.

We apply sparse logistic regression (LR), support vector machines (SVM), diagonal linear discriminant analysis (DLDA), and diagonal quadratic discriminant analysis (DQDA) on this dataset. For each method, we conduct 100 runs, within each run, 40% of the data are randomly selected as training and the remaining 60% are used for testing.

Figure 14 illustrates the regularization paths of sparse logistic regression using the time-domain and frequency-domain features. A regularization path provides the coefficient value of each feature over all regularization parameters. Since the time-domain features are highly correlated, the corresponding regularization path is quite irregular. In contrast, the paths for the frequency-domain features behave stably.

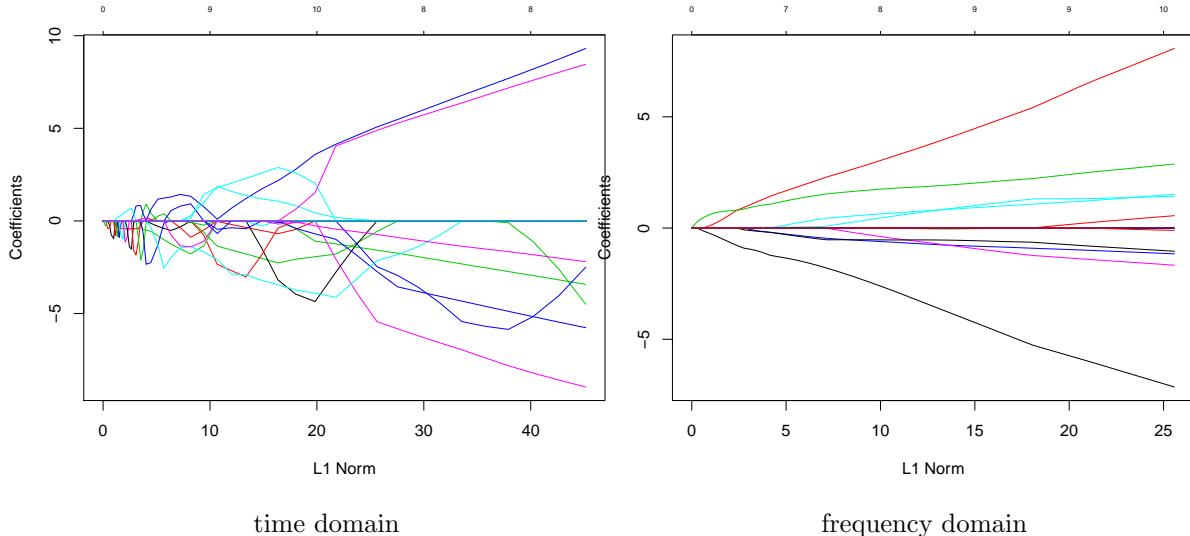


Figure 14: The regularization paths of sparse logistic regression using the features of time-domain and frequency-domain. The vertical axis corresponds to the values of the coefficients, plotted as a function of their ℓ_1 -norm. The path using time-domain features are highly irregular, while the path using frequency-domain features are more stable.

Figure 15 compares the classification performance of all these methods. The results show that classification in the frequency domain is not helpful. The regularization paths of the SVM are the same in both the time and frequency domains. This is expected since the discrete cosine transformation is an orthonormal transformation, which corresponds to rotating the data in the feature space while preserving their Euclidean distances and inner products. It is easy

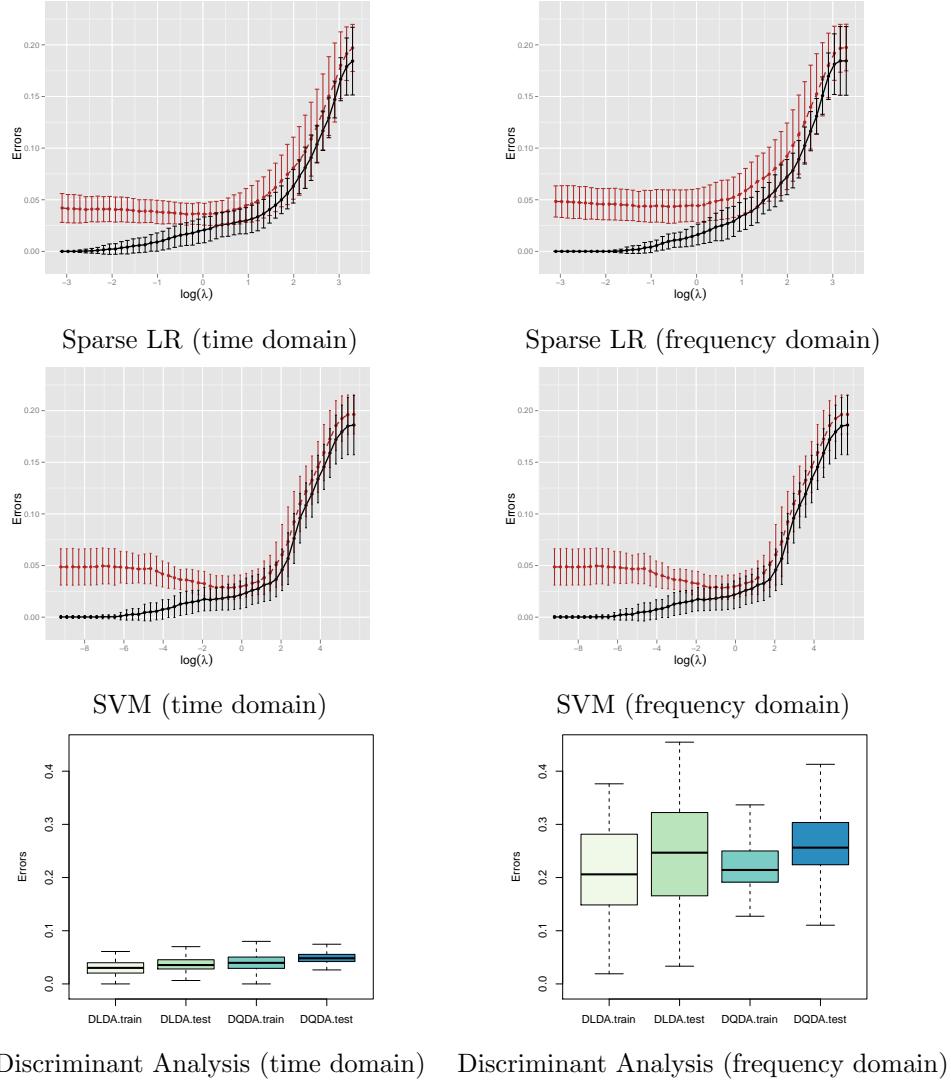


Figure 15: Comparison of different methods on the supernova dataset using both the time-domain (left column) and frequency-domain features (right Column). Top four figures: mean error curves (black: training error, red: test error) and their corresponding standard error bars for sparse logistic regression (LR) and support vector machines (SVM). Bottom two figures: boxplots of the training and test errors of diagonal linear discriminant analysis (DLDA) and diagonal quadratic discriminant analysis (DQDA). For the time-domain features, the SVM achieves the smallest test error among all methods.

to see that the SVM is rotation invariant. Sparse logistic regression is not rotation invariant due to the ℓ_1 -norm regularization term. The performance of the sparse logistic regression in the frequency domain is worse than that in the time domain. The DLDA and DQDA are also not rotation invariant; their performances decreases significantly in the frequency domain compared to those in the time domain. In both time and frequency domains, the SVM outperforms all the other methods. Then follows sparse logistic regression, which is better than DLDA and DQDA.

10 Case Study II: Political Blog Classification

In this example, we classify political blogs according to whether their political leanings are *liberal* or *conservative*. Snapshots of two political blogs are shown in Figure 16.



Figure 16: Examples of two political blogs with different orientations, one conservative and the other liberal.

The data consist of 403 political blogs in a two-month window before the 2004 presidential election. Among these blogs, 205 are *liberal* and 198 are *conservative*. We use *bag-of-words* features, i.e., each unique word from these 403 blogs serves as a feature. For each blog, the value of a feature is the number of occurrences of the word normalized by the total number of words in the blog. After converting all words to lower case, we remove stop words and only retain words with at least 10 occurrences across all the 403 blogs. This results in 23,955 features, each of which corresponds to an English word. Such features are only a crude representation of the text represented as an unordered collection of words, disregarding all grammatical structure. We also extracted features that use hyperlink information. In particular, we selected 292 out of the 403 blogs that are heavily linked to, and for each blog $i = 1, \dots, 403$, its linkage information is represented as a 292-dimensional binary vector $(x_{i1}, \dots, x_{i292})^T$ where $x_{ij} = 1$ if the i th blog has a link to the j th feature blog. The total number of covariates is then $23,955 + 292 = 24,247$. Even though the link features only constitute a small proportion, they are important for predictive accuracy.

We run the full regularization paths of sparse logistic regression and support vector machines,

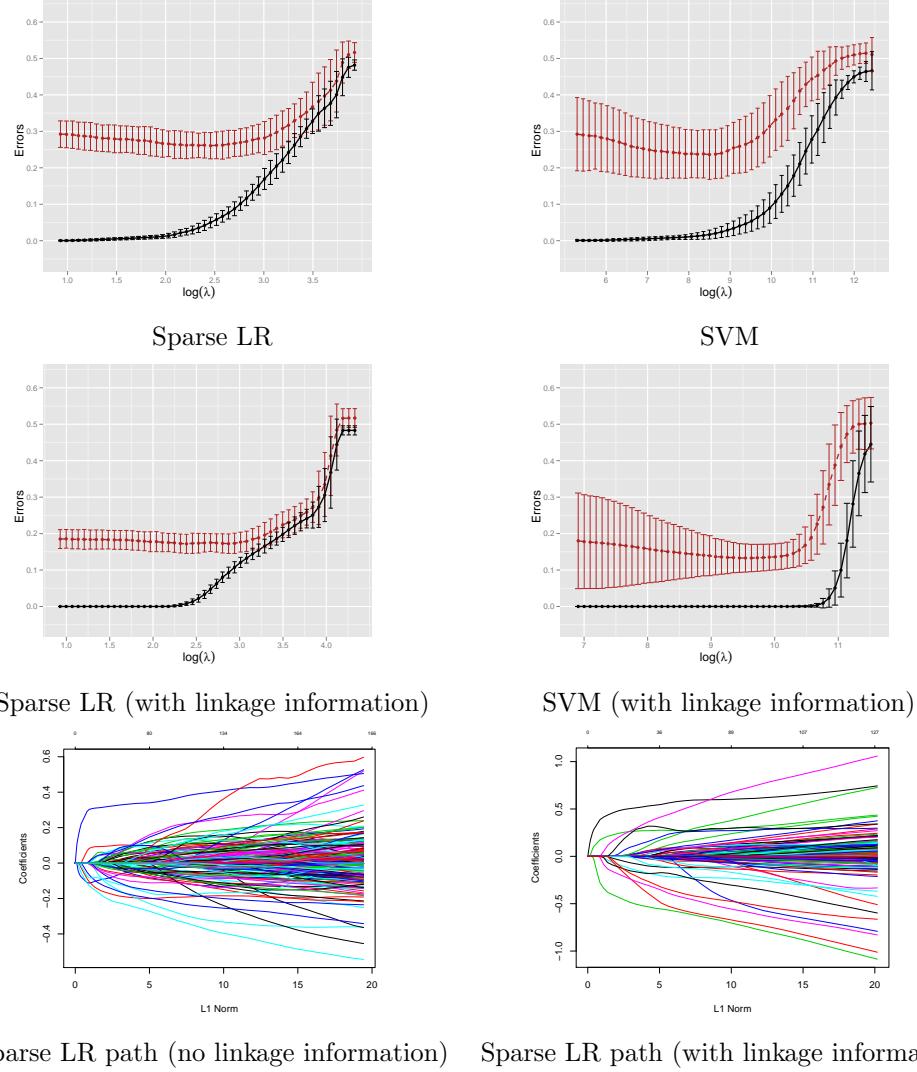


Figure 17: Comparison of the sparse logistic regression (LR) and support vector machine (SVM) on the political blog data. (Top four figures): The mean error curves (Black: training error, Red: test error) and their corresponding standard error bars of the sparse LR and SVM, with and without the linkage information. (Bottom two figures): Two typical regularization paths of the sparse logistic LR with and without the linkage information. On this dataset, the diagonal linear discriminant analysis (DLDA) achieves a test error 0.303 ($sd = 0.07$) without the linkage information and a test error 0.159 ($sd = 0.02$) with the linkage information.

100 times each. For each run, the data are randomly partitioned into training (60%) and testing (40%) sets. Figure 17 shows the mean error curves with their standard errors. From Figure 17, we see that linkage information is crucial. Without the linkage features, the smallest mean test error of the support vector machine along the regularization path is 0.247, while that of the sparse logistic regression is 0.270. With the link features, the smallest test error for the support vector machine becomes 0.132. Although the support vector machine has a better mean error curve, it has much larger standard error. Two typical regularization paths for sparse logistic regression with and without using the link features are provided at the bottom of Figure 17. By examining these paths, we see that when the link features are used, 11 of the first 20 selected features are link features. In this case, although the class conditional distribution is obviously not Gaussian, we still apply the diagonal linear discriminant analysis (DLDA) on this dataset for a comparative study. Without the linkage features, the DLDA has a mean test error 0.303 ($sd = 0.07$). With the linkage features, DLDA has a mean test error 0.159 ($sd = 0.02$).

Nonparametric Classification

10/36-702

1 Introduction

Let $h : \mathcal{X} \rightarrow \{0, 1\}$ to denote a classifier where \mathcal{X} is the domain of X . In parametric classification we assumed that h took a very constrained form, typically linear. In nonparametric classification we aim to relax this assumption.

Let us recall a few definitions and facts. The *classification risk*, or *error rate*, of h is

$$R(h) = \mathbb{P}(Y \neq h(X)) \quad (1)$$

and the *empirical error rate* or *training error rate* based on training data $(X_1, Y_1), \dots, (X_n, Y_n)$ is

$$\widehat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n I(h(X_i) \neq Y_i). \quad (2)$$

$R(h)$ is minimized by the *Bayes' rule*

$$h^*(x) = \begin{cases} 1 & \text{if } m(x) > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} = \begin{cases} 1 & \text{if } \frac{p_1(x)}{p_0(x)} > \frac{(1-\pi)}{\pi} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where $m(x) = \mathbb{P}(Y = 1 | X = x)$, $p_j(x) = p(x | Y = j)$ and $\pi = \mathbb{P}(Y = 1)$. The *excess risk* of a classifier h is $R(h) - R(h^*)$.

In the multiclass case, $Y \in \{1, \dots, k\}$, the Bayes' rule is

$$h^*(x) = \operatorname{argmax}_{1 \leq j \leq k} \pi_j p_j(x) = \operatorname{argmax}_{1 \leq j \leq k} m_j(x)$$

where $m_j(x) = \mathbb{P}(Y = j | X = x)$, $\pi_j = \mathbb{P}(Y = j)$ and $p_j(x) = p(x | Y = j)$.

2 Plugin Methods

One approach to nonparametric classification is to estimate the unknown quantities in the expression for the Bayes' rule (3) and simply plug them in. For example, if \widehat{m} is any nonparametric regression estimator then we can use

$$\widehat{h}(x) = \begin{cases} 1 & \text{if } \widehat{m}(x) > \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

For example, we could use the kernel regression estimator

$$\widehat{m}_h(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{\|x-X_i\|}{h}\right)}{\sum_{i=1}^n K\left(\frac{\|x-X_i\|}{h}\right)}.$$

However, the bandwidth should be optimized for classification error as described in Section 8.

We have the following theorem.

Theorem 1 *Let \widehat{h} be the plug-in classifier based on \widehat{m} . Then,*

$$R(\widehat{h}) - R(h^*) \leq 2 \int |\widehat{m}(x) - m(x)| dP(x) \leq 2 \sqrt{\int |\widehat{m}(x) - m(x)|^2 dP(x)}. \quad (5)$$

An immediate consequence of this theorem is that any result about nonparametric regression can be turned into a result about nonparametric classification. For example, if $\int |\widehat{m}(x) - m(x)|^2 dP(x) = O_P(n^{-2\beta/(2\beta+d)})$ then $R(\widehat{h}) - R(h^*) = O_P(n^{-\beta/(2\beta+d)})$. However, (5) is an upper bound and it is possible that $R(\widehat{h}) - R(h^*)$ is strictly smaller than $\sqrt{\int |\widehat{m}(x) - m(x)|^2 dP(x)}$.

When $Y \in \{1, \dots, k\}$ the plugin rule has the form

$$\widehat{h}(x) = \operatorname{argmax}_j \widehat{m}_j(x)$$

where $\widehat{m}_j(x)$ is an estimate of $\mathbb{P}(Y = j | X = x)$.

3 Classifiers Based on Density Estimation

We can apply nonparametric density estimation to each class to get estimators \widehat{p}_0 and \widehat{p}_1 . Then we define

$$\widehat{h}(x) = \begin{cases} 1 & \text{if } \frac{\widehat{p}_1(x)}{\widehat{p}_0(x)} > \frac{(1-\widehat{\pi})}{\widehat{\pi}} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $\widehat{\pi} = n^{-1} \sum_{i=1}^n Y_i$. Hence, any nonparametric density estimation method yields a nonparametric classifier.

A simplification occurs if we assume that the covariate has independent coordinates, conditioned on the class variable Y . Thus, if $X_i = (X_{i1}, \dots, X_{id})^T$ has dimension d and if we assume conditional independence, then the density factors as $p_j(x) = \prod_{\ell=1}^d p_{j\ell}(x_\ell)$. In

this case we can estimate the one-dimensional marginals $p_{j\ell}(x_\ell)$ separately and then define $\widehat{p}_j(x) = \prod_{\ell=1}^d \widehat{p}_{j\ell}(x_\ell)$. This has the advantage that we never have to do more than a one-dimensional density estimate. This approach is called *naive Bayes*. The resulting classifier can sometimes be very accurate even if the independence assumption is false.

It is easy to extend density based methods for multiclass problems. If $Y \in \{1, \dots, k\}$ then we estimate the k densities $\widehat{p}_j(x) = p(x|Y=j)$ and the classifier is

$$\widehat{h}(x) = \operatorname{argmax}_j \widehat{\pi}_j \widehat{p}_j(x)$$

where $\widehat{\pi}_j = n^{-1} \sum_{i=1}^n I(Y_i = j)$.

4 Nearest Neighbors

The *k-nearest neighbor classifier* is

$$h(x) = \begin{cases} 1 & \sum_{i=1}^n w_i(x) I(Y_i = 1) > \sum_{i=1}^n w_i(x) I(Y_i = 0) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $w_i(x) = 1$ if X_i is one of the k nearest neighbors of x , $w_i(x) = 0$, otherwise. “Nearest” depends on how you define the distance. Often we use Euclidean distance $\|X_i - X_j\|$. In that case you should standardize the variables first.

The *k-nearest neighbor classifier* can be recast as a plugin rule. Define the regression estimator

$$\widehat{m}(x) = \frac{\sum_{i=1}^n Y_i I(\|X_i - x\| \leq d_k(x))}{\sum_{i=1}^n I(\|X_i - x\| \leq d_k(x))}$$

where $d_k(x)$ is the distance between x and its k^{th} -nearest neighbor. Then $\widehat{h}(x) = I(\widehat{m}(x) > 1/2)$.

It is interesting to consider the classification error when n is large. First suppose that $k = 1$ and consider a fixed x . Then $\widehat{h}(x)$ is 1 if the closest X_i has label $Y = 1$ and $\widehat{h}(x)$ is 0 if the closest X_i has label $Y = 0$. When n is large, the closest X_i is approximately equal to x . So the probability of an error is

$$m(X_i)(1-m(x)) + (1-m(X_i))m(x) \approx m(x)(1-m(x)) + (1-m(x))m(x) = 2m(x)(1-m(x)).$$

Define

$$L_n = \mathbb{P}(Y \neq \widehat{h}(X) | D_n)$$

where $D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$. Then we have that

$$\lim_{n \rightarrow \infty} \mathbb{E}(L_n) = \mathbb{E}(2m(X)(1-m(X))) \equiv R_{(1)}. \quad (8)$$

The Bayes risk can be written as $R_* = \mathbb{E}(A)$ where $A = \min\{m(X), 1 - m(X)\}$. Note that $A \leq 2m(X)(1 - m(X))$. Also, by direct integration, $\mathbb{E}(A(1 - A)) \leq \mathbb{E}(A)\mathbb{E}(1 - A)$. Hence, we have the well-known result due to Cover and Hart (1967),

$$R_* \leq R_{(1)} = \mathbb{E}(A(1 - A)) \leq 2\mathbb{E}(A)\mathbb{E}(1 - A) = 2R_*(1 - R_*) \leq 2R_*.$$

Thus, for any problem with small Bayes error, $k = 1$ nearest neighbors should have small error.

More generally, for any odd k ,

$$\lim_{n \rightarrow \infty} \mathbb{E}(L_n) = R_{(k)} \quad (9)$$

where

$$R_{(k)} = \mathbb{E} \left(\sum_{j=0}^k \binom{k}{j} m^j(X)(1 - m(X))^{k-j} [m(X)I(j < k/2) + (1 - m(X))I(j > k/2)] \right).$$

Theorem 2 (Devroye et al 1996) *For all odd k ,*

$$R_* \leq R_{(k)} \leq R_* + \frac{1}{\sqrt{ke}}. \quad (10)$$

Proof. We can rewrite $R_{(k)}$ as $R_{(k)} = \mathbb{E}(a(m(X)))$ where

$$a(z) = \min\{z, 1 - z\} + |2z - 1| \mathbb{P}\left(B > \frac{k}{2}\right)$$

and $B \sim \text{Binomial}(k, \min\{z, 1 - z\})$. The mean of $a(z)$ is less than or equal to its maximum and, by symmetry, we can take the maximum over $0 \leq z \leq 1/2$. Hence, letting $B \sim \text{Binomial}(k, z)$, we have, by Hoeffding's inequality,

$$R_{(k)} - R_* \leq \sup_{0 \leq z \leq 1/2} (1 - 2z) \mathbb{P}\left(B > \frac{k}{2}\right) \leq \sup_{0 \leq z \leq 1/2} (1 - 2z)e^{-2k(1/2-z)^2} = \sup_{0 \leq u \leq 1} ue^{-ku^2/2} = \frac{1}{\sqrt{ke}}.$$

□

If the distribution of X has a density function then we have the following.

Theorem 3 (Devroye and Györfi 1985) *Suppose that the distribution of X has a density and that $k \rightarrow \infty$ and $k/n \rightarrow 0$. For every $\epsilon > 0$ the following is true. For all large n ,*

$$\mathbb{P}(R(\hat{h}) - R_* > \epsilon) \leq e^{-n\epsilon^2/(72\gamma_d^2)}$$

where \hat{h}_n is the k -nearest neighbor classifier estimated on a sample of size n , and where γ_d depends on the dimension d of X .

Recently, Chaudhuri and Dasgupta (2014) have obtained some very general results about k-nn classifiers. We state one of their key results here.

Theorem 4 (Chaudhuri and Dasgupta 2014) *Suppose that*

$$P(\{x : |m(x) - (1/2)| \leq t\}) \leq Ct^\beta$$

for some $\beta \geq 0$ and some $C > 0$. Also, suppose that m satisfies the following smoothness condition: for all x and $r > 0$

$$|m(B) - m(x)| \leq LP(B^o)^\alpha$$

where $B = \{u : \|x-u\| \leq r\}$, $B^o = \{u : \|x-u\| < r\}$ and $m(B) = (P(B))^{-1} \int_B m(u)dP(x)$. Fix any $0 < \delta < 1$. Let h_* be the Bayes rule. With probability at least $1 - \delta$,

$$P(\hat{h}(X) \leq h_*(X)) \leq \delta C \left(\frac{\log(1/\delta)}{n} \right)^{\frac{\alpha\beta}{2\alpha+1}}.$$

If $k \asymp n^{\frac{2\alpha}{2\alpha+1}}$ then

$$R(\hat{h}) - R(h_*) \preceq n^{-\frac{\alpha(\beta+1)}{2\alpha+1}}.$$

4.1 Partitions and Trees

As with nonparametric regression, simple and interpretable classifiers can be derived by partitioning the range of X . Let $\Pi_n = \{A_1, \dots, A_N\}$ be a partition of \mathcal{X} . Let A_j be the partition element that contains x . Then $\hat{h}(x) = 1$ if $\sum_{X_i \in A_j} Y_i \geq \sum_{X_i \in A_j} (1 - Y_i)$ and $\hat{h}(x) = 0$ otherwise. This is nothing other than the plugin classifier based on the partition regression estimator

$$\hat{m}(x) = \sum_{j=1}^N \bar{Y}_j I(x \in A_j)$$

where $\bar{Y}_j = n_j^{-1} \sum_{i=1}^n Y_i I(X_i \in A_j)$ is the average of the Y_i 's in A_j and $n_j = \#\{X_i \in A_j\}$. (We define \bar{Y}_j to be 0 if $n_j = 0$.)

Recall from the results on regression that if

$$m \in \mathcal{M} = \left\{ m : |m(x) - m(z)| \leq L\|x - z\|, \quad x, z \in \mathbb{R}^d \right\} \quad (11)$$

and the binwidth b satsfies $b \asymp n^{-1/(d+2)}$ then

$$\mathbb{E}\|\hat{m} - m\|_P^2 \leq \frac{c}{n^{2/(d+2)}}. \quad (12)$$

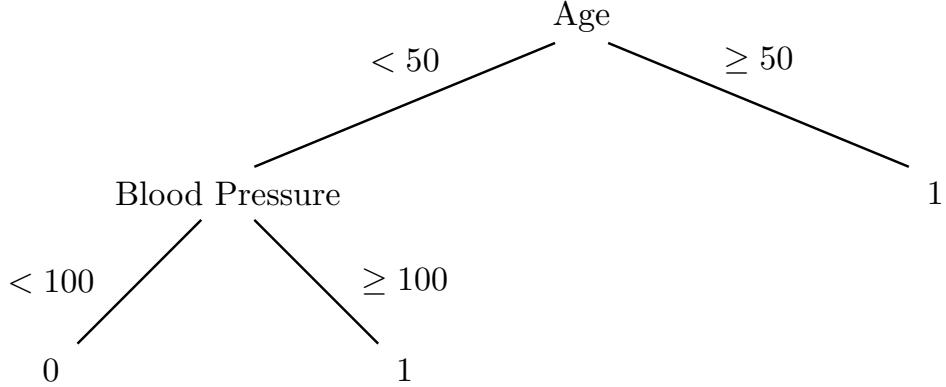


Figure 1: A simple classification tree.

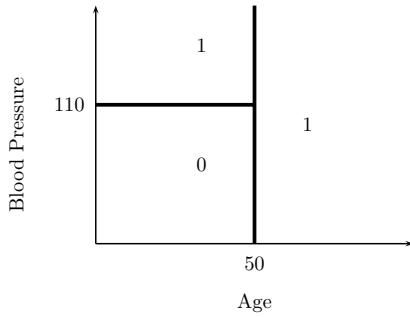


Figure 2: Partition representation of classification tree.

From (5), we conclude that $R(\hat{h}) - R(h_*) = O(n^{-1/(d+2)})$. However, this binwidth was based on the bias-variance tradeoff of the regression problem. For classification, b should be chosen as described in Section 8.

Like regression trees, *classification trees* are partition classifiers where the partition is built recursively. For illustration, suppose there are two covariates, $X_1 = \text{age}$ and $X_2 = \text{blood pressure}$. Figure 1 shows a classification tree using these variables.

The tree is used in the following way. If a subject has $\text{Age} \geq 50$ then we classify him as $Y = 1$. If a subject has $\text{Age} < 50$ then we check his blood pressure. If systolic blood pressure is < 100 then we classify him as $Y = 1$, otherwise we classify him as $Y = 0$. Figure 2 shows the same classifier as a partition of the covariate space.

Here is how a tree is constructed. First, suppose that $y \in \mathcal{Y} = \{0, 1\}$ and that there is only a single covariate X . We choose a split point t that divides the real line into two sets

$A_1 = (-\infty, t]$ and $A_2 = (t, \infty)$. Let $r_s(j)$ be the proportion of observations in A_s such that $Y_i = j$:

$$r_s(j) = \frac{\sum_{i=1}^n I(Y_i = j, X_i \in A_s)}{\sum_{i=1}^n I(X_i \in A_s)} \quad (13)$$

for $s = 1, 2$ and $j = 0, 1$. The *impurity* of the split t is defined to be $I(t) = \sum_{s=1}^2 \gamma_s$ where

$$\gamma_s = 1 - \sum_{j=0}^1 r_s(j)^2. \quad (14)$$

This particular measure of impurity is known as the *Gini index*. If a partition element A_s contains all 0's or all 1's, then $\gamma_s = 0$. Otherwise, $\gamma_s > 0$. We choose the split point t to minimize the impurity. Other indices of impurity besides the Gini index can be used, such as entropy. The reason for using impurity rather than classification error is because impurity is a smooth function and hence is easy to minimize.

When there are several covariates, we choose whichever covariate and split that leads to the lowest impurity. This process is continued until some stopping criterion is met. For example, we might stop when every partition element has fewer than n_0 data points, where n_0 is some fixed number. The bottom nodes of the tree are called the *leaves*. Each leaf is assigned a 0 or 1 depending on whether there are more data points with $Y = 0$ or $Y = 1$ in that partition element.

This procedure is easily generalized to the case where $Y \in \{1, \dots, K\}$. We define the impurity by

$$\gamma_s = 1 - \sum_{j=1}^k r_s(j)^2 \quad (15)$$

where $r_i(j)$ is the proportion of observations in the partition element for which $Y = j$.

5 Minimax Results

The minimax classification risk over a set of joint distributions \mathcal{P} is

$$R_n(\mathcal{P}) = \inf_{\hat{h}} \sup_{P \in \mathcal{P}} \left(R(\hat{h}) - R_n^* \right) \quad (16)$$

where $R(\hat{h}) = \mathbb{P}(Y \neq \hat{h}(X))$, R_n^* is the Bayes error and the infimum is over all classifiers constructed from the data $(X_1, Y_1), \dots, (X_n, Y_n)$. Recall that

$$R(\hat{h}) - R(h^*) \leq 2 \sqrt{\int |\hat{m}(x) - m(x)|^2 dP(x)}$$

Class	Rate	Condition
$\mathcal{E}(\alpha)$	$n^{-\alpha/(2\alpha+d)}$	$\alpha > 1/2$
BV	$n^{-1/3}$	
MI	$\sqrt{\log n/n}$	
$L(\alpha, q)$	$n^{-\alpha/(2\alpha+1)}$	$\alpha > (1/q - 1/2)_+$
$B_{\sigma,q}^\alpha$	$n^{-\alpha/(2\alpha+d)}$	$\alpha/d > 1/q - 1/2$
Neural nets	see text	

Table 1: Minimax Rates of Convergence for Classification.

Thus $R_n(\mathcal{P}) \leq 2\sqrt{\tilde{R}_n(\mathcal{P})}$ where $\tilde{R}_n(\mathcal{P})$ is the minimax risk for estimating the regression function m . Since this is just an inequality, it leaves open the following question: can $R_n(\mathcal{P})$ be substantially smaller than $2\sqrt{\tilde{R}_n(\mathcal{P})}$? Yang (1999) proved that the answer is no, in cases where \mathcal{P} is substantially rich. Moreover, we can achieve minimax classification rates using plugin regression methods.

However, with smaller classes that invoke extra assumptions, such as the *Tsybakov noise condition*, there can be a dramatic difference. Here, we summarize Yang's results under the richness assumption. This assumption is simply that if m is in the class, then a small hypercube containing m is also in the class. Yang's results are summarized in Table 1.

The classes in Table 1 are the following: $\mathcal{E}(\alpha)$ is the Sobolev space of order α , BV is the class of functions of bounded variation, MI is all monotone functions, $L(\alpha, q)$ are α -Lipschitz (in q -norm), and $B_{\sigma,q}^\alpha$ are Besov spaces. For neural nets we have the bound, for every $\epsilon > 0$,

$$\left(\frac{1}{n}\right)^{\frac{1+(2/d)}{4+(4/d)}+\epsilon} \leq R_n(\mathcal{P}) \leq \left(\frac{\log n}{n}\right)^{\frac{1+(1/d)}{4+(2/d)}}$$

It appears that, as $d \rightarrow \infty$, we get the dimension independent rate $(\log n/n)^{1/4}$. However, this result requires some caution since the class of distributions implicitly gets smaller as d increases.

6 Support Vector Machines

When we discussed linear classification, we defined SVM classifier $\hat{h}(x) = \text{sign}(\hat{H}(x))$ where $\hat{H}(x) = \hat{\beta}_0 + \hat{\beta}^T x$ and $\hat{\beta}$ minimizes

$$\sum_i [1 - Y_i H(X_i)]_+ + \lambda \|\beta\|_2^2.$$

We can do a nonparametric version by letting H be in a RKHS and taking the penalty to be $\|H\|_K^2$. In terms of implementation, this means replacing every instance of an inner product $\langle X_i, X_j \rangle$ with $K(X_i, X_j)$.

7 Boosting

Boosting refers to a class of methods that build classifiers in a greedy, iterative way. The original boosting algorithm is called *AdaBoost* and is due to Freund and Schapire (1996). See Figure 3.

The algorithm seems mysterious and there is quite a bit of controversy about why (and when) it works. Perhaps the most compelling explanation is due to Friedman, Hastie and Tibshirani (2000) which is the explanation we will give. However, the reader is warned that there is not consensus on the issue. Further discussions can be found in Bühlmann and Hothorn (2007), Zhang and Yu (2005) and Mease and Wyner (2008). The latter paper is followed by a spirited discussion from several authors. Our view is that boosting combines two distinct ideas: *surrogate loss functions* and *greedy function approximation*.

In this section, we assume that $Y_i \in \{-1, +1\}$. Many classifiers then have the form

$$h(x) = \text{sign}(H(x))$$

for some function $H(x)$. For example, a linear classifier corresponds to $H(x) = \beta^T x$. The risk can then be written as

$$R(h) = \mathbb{P}(Y \neq h(X)) = \mathbb{P}(YH(X) < 0) = \mathbb{E}(L(A))$$

where $A = YH(X)$ and $L(a) = I(a < 0)$. As a function of a , the loss $L(a)$ is discontinuous which makes it difficult to work with. Friedman, Hastie and Tibshirani (2000) show that AdaBoost corresponds to using a surrogate loss, namely, $L(a) = e^{-a} = e^{-yH(x)}$. Consider finding a classifier of the form $\sum_m \alpha_m h_m(x)$ by minimizing the exponential loss $\sum_i e^{-Y_i H(X_i)}$. If we do this iteratively, adding one function at a time, this leads precisely to AdaBoost. Typically, the classifiers h_j in the sum $\sum_m \alpha_m h_m(x)$ are taken to be very simple classifiers such as small classification trees.

The argument in Friedman, Hastie and Tibshirani (2000) is as follows. Consider minimizing the expected loss $J(F) = \mathbb{E}(e^{-YF(X)})$. Suppose our current estimate is F and consider updating to an improved estimate $F(x) + cf(x)$. Expanding around $f(x) = 0$,

$$\begin{aligned} J(F + cf) &= \mathbb{E}(e^{-Y(F(X)+cf(X))}) \approx \mathbb{E}(e^{-YF(X)}(1 - cYf(X) + c^2Y^2f^2(X)/2)) \\ &= \mathbb{E}(e^{-YF(X)}(1 - cYf(X) + c^2/2)) \end{aligned}$$

since $Y^2 = f^2(X) = 1$. Now consider minimizing the latter expression a fixed $X = x$. If we minimize over $f(x) \in \{-1, +1\}$ we get $f(x) = 1$ if $E_w(y|x) > 0$ and $f(x) = -1$ if

1. Input: $(X_1, Y_1), \dots, (X_n, Y_n)$ where $Y_i \in \{-1, +1\}$.
2. Set $w_i = 1/n$ for $i = 1, \dots, n$.
3. Repeat for $m = 1, \dots, M$.
 - (a) Compute the weighted error $\epsilon(h) = \sum_{i=1}^n w_i I(Y_i \neq h(X_i))$ and find h_m to minimize $\epsilon(h)$.
 - (b) Let $\alpha_m = (1/2) \log((1 - \epsilon)/\epsilon)$.
 - (c) Update the weights:
$$w_i \leftarrow \frac{w_i e^{-\alpha_m Y_i h_m(X_i)}}{Z}$$

where Z is chosen so that the weights sum to 1.
4. The final classifier is

$$h(x) = \text{sign} \left(\sum_{m=1}^M \alpha_m h_m(x) \right).$$

Figure 3: AdaBoost

$E_w(y|x) < 0$ where $E_w(y|x) = E(w(x,y)y|x)/E(w(x,y)|x)$ and $w(x,y) = e^{-yF(x)}$. In other words, the optimal f is simply the Bayes classifier with respect to the weights. This is exactly the first step in AdaBoost. If we fix now fix $f(x)$ and minimize over c we get

$$c = \frac{1}{2} \log \left(\frac{1-\epsilon}{\epsilon} \right)$$

where $\epsilon = E_w(I(Y \neq f(x)))$. Thus the updated $F(x)$ is

$$F(x) \leftarrow F(x) + cf(x)$$

as in AdaBoost. When we update F this way, we change the weights to

$$w(x,y) \leftarrow w(x,y)e^{-cf(x)y} = w(x,y) \exp \left(\log \left(\frac{1-\epsilon}{\epsilon} \right) I(Y \neq f(x)) \right)$$

which again is the same as AadBoost.

Seen in this light, boosting really combines two ideas. The first is the use of surrogate loss functions. The second is greedy function approximation.

8 Choosing Tuning Parameters

All the nonparametric methods involve tuning parameters, for example, the number of neighbors k in nearest neighbors. As with density estimation and regression, these parameters can be chosen by a variety of cross-validation methods. Here we describe the *data splitting* version of cross-validation. Suppose the data are $(X_1, Y_1), \dots, (X_{2n}, Y_{2n})$. Now randomly split the data into two halves that we denote by

$$\mathcal{D} = \left\{ (\tilde{X}_1, \tilde{Y}_1), \dots, (\tilde{X}_n, \tilde{Y}_n) \right\}, \quad \text{and} \quad \mathcal{E} = \left\{ (X_1^*, Y_1^*), \dots, (X_n^*, Y_n^*) \right\}.$$

Construct classifiers $\mathcal{H} = \{h_1, \dots, h_N\}$ from \mathcal{D} corresponding to different values of the tuning parameter. Define the risk estimator

$$\widehat{R}(h_j) = \frac{1}{n} \sum_{i=1}^n I(Y_i^* \neq h_j(X_i^*)).$$

Let $\widehat{h} = \operatorname{argmin}_{h \in \mathcal{H}} \widehat{R}(h)$.

Theorem 5 *Let $h_* \in \mathcal{H}$ minimize $R(h) = \mathbb{P}(Y \neq h(X))$. Then*

$$\mathbb{P} \left(R(\widehat{h}) > R(h_*) + 2 \sqrt{\frac{1}{2n} \log \left(\frac{2N}{\delta} \right)} \right) \leq \delta.$$

Proof. By Hoeffding's inequality, $\mathbb{P}(|\widehat{R}(h) - R(h)| > \epsilon) \leq 2e^{-2n\epsilon^2}$, for each $h \in \mathcal{H}$. By the union bound,

$$\mathbb{P}(\max_{h \in \mathcal{H}} |\widehat{R}(h) - R(h)| > \epsilon) \leq 2Ne^{-2n\epsilon^2} = \delta$$

where $\epsilon = \sqrt{\frac{1}{2n} \log \left(\frac{2N}{\delta} \right)}$. Hence, except on a set of probability at most δ ,

$$R(\widehat{h}) \leq \widehat{R}(\widehat{h}) + \epsilon \leq \widehat{R}(\widehat{h}_*) + \epsilon \leq R(\widehat{h}_*) + 2\epsilon.$$

□

Note that the difference between $R(\widehat{h})$ and $R(h_*)$ is $O(\sqrt{\log N/n})$ but in regression it was $O(\log N/n)$ which is an interesting difference between the two settings. Under low noise conditions, the error can be improved.

A popular modification of data-splitting is *K-fold cross-validation*. The data are divided into K blocks; typically $K = 10$. One block is held out as test data to estimate risk. The process is then repeated K times, leaving out a different block each time, and the results are averaged over the K repetitions.

9 Example

The following data are from simulated images of gamma ray events for the Major Atmospheric Gamma-ray Imaging Cherenkov Telescope (MAGIC) in the Canary Islands. The data are from archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope. The telescope studies gamma ray bursts, active galactic nuclei and supernovae remnants. The goal is to predict if an event is real or is background (hadronic shower). There are 11 predictors that are numerical summaries of the images. We randomly selected 400 training points (200 positive and 200 negative) and 1000 test cases (500 positive and 500 negative). The results of various methods are in Table 2. See Figures 4, 5, 6, 7.

10 Sparse Nonparametric Logistic Regression

For high dimensional problems we can use sparsity-based methods. The nonparametric additive logistic model is

$$\mathbb{P}(Y = 1 | X) \equiv p(X; f) = \frac{\exp \left(\sum_{j=1}^p f_j(X_j) \right)}{1 + \exp \left(\sum_{j=1}^p f_j(X_j) \right)} \quad (17)$$

where $Y \in \{0, 1\}$, and the population log-likelihood is

$$\ell(f) = \mathbb{E} [Yf(X) - \log(1 + \exp f(X))] \quad (18)$$

Method	Test Error
Logistic regression	0.23
SVM (Gaussian Kernel)	0.20
Kernel Regression	0.24
Additive Model	0.20
Reduced Additive Model	0.20
11-NN	0.25
Trees	0.20

Table 2: Various methods on the MAGIC data. The reduced additive model is based on using the three most significant variables from the additive model.

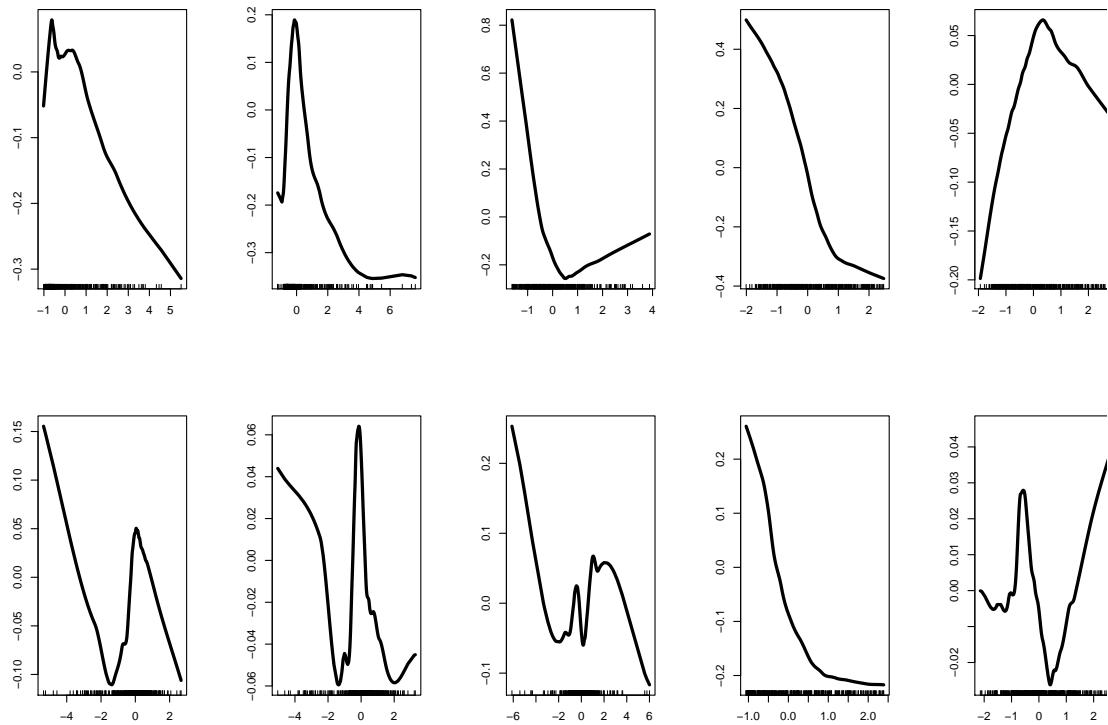


Figure 4: Estimated functions for additive model.

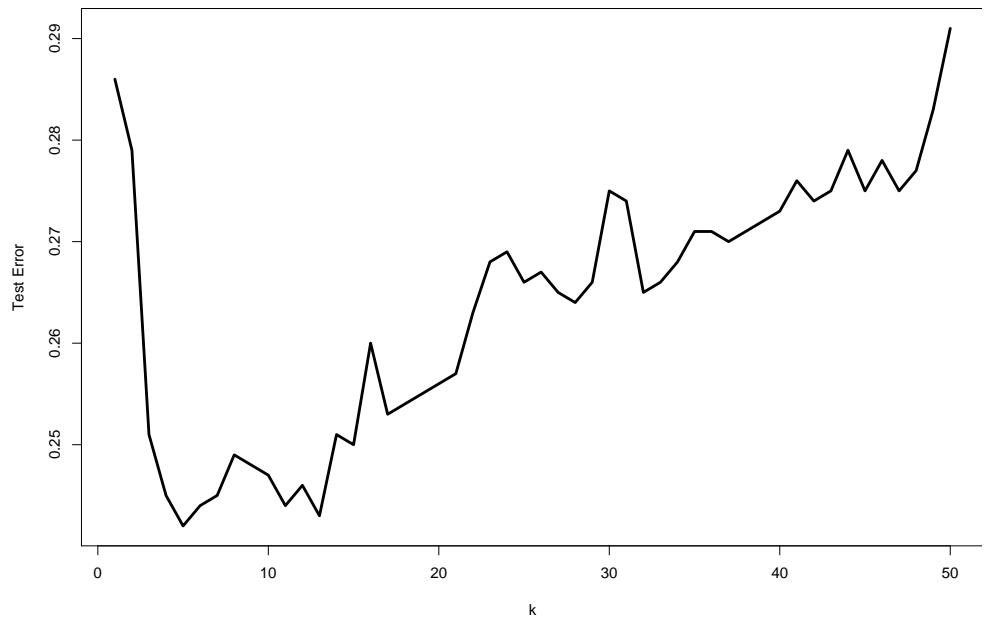


Figure 5: Test error versus k for nearest neighbor estimator.

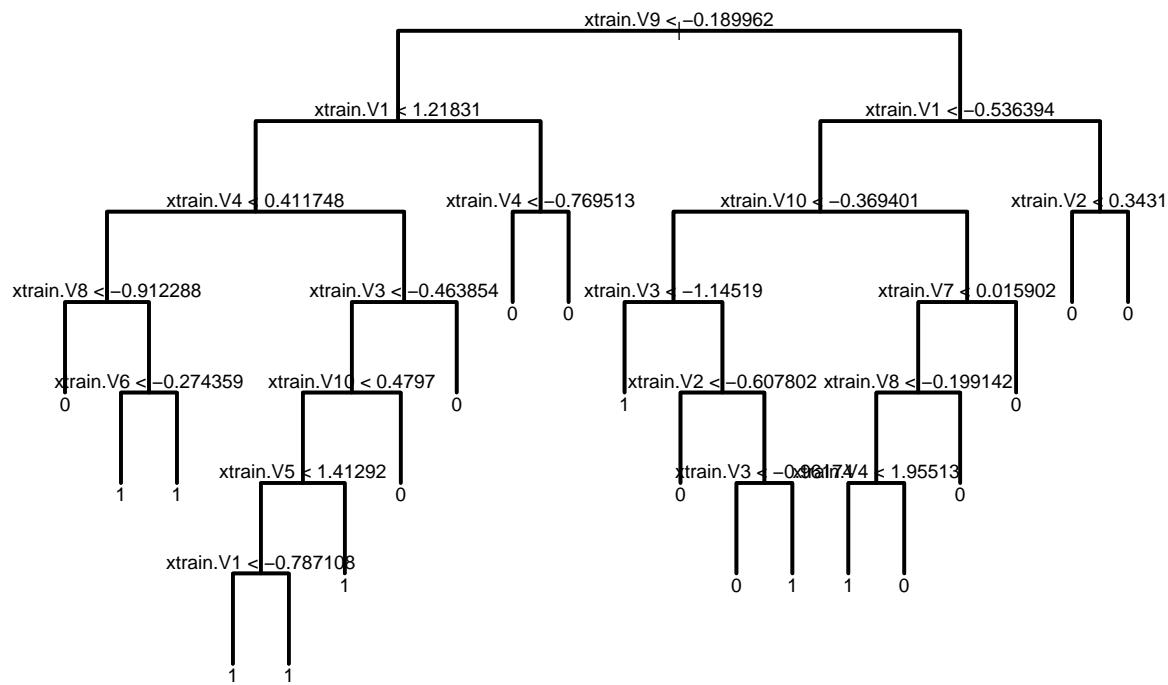


Figure 6: Full tree.

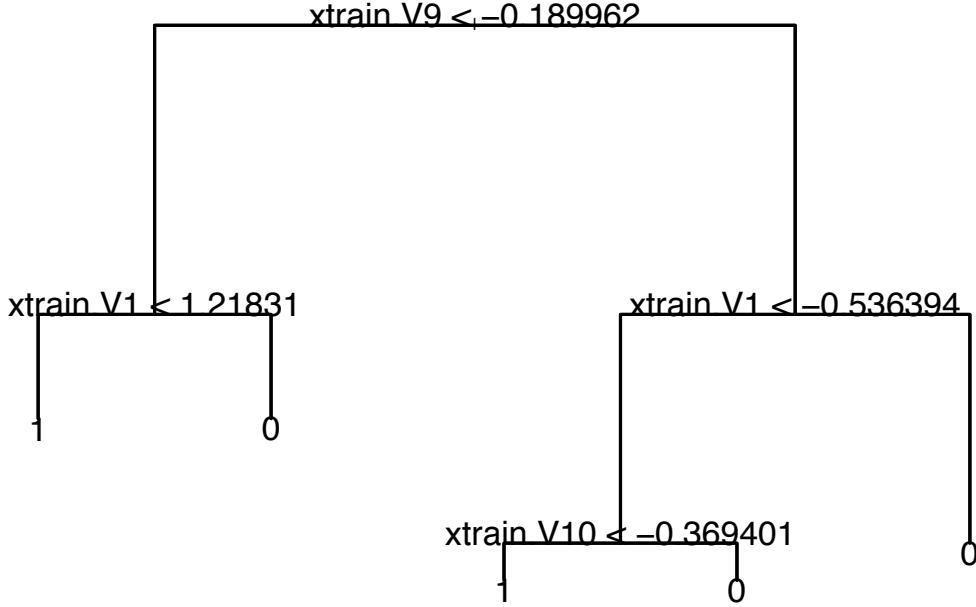


Figure 7: Classification tree. The size of the tree was chosen by cross-validation.

where $f(X) = \sum_{j=1}^p f_j(X_j)$. To fit this model, the local scoring algorithm runs the backfitting procedure within Newton's method. One iteratively computes the transformed response for the current estimate \hat{f}

$$Z_i = \hat{f}(X_i) + \frac{Y_i - p(X_i; \hat{f})}{p(X_i; \hat{f})(1 - p(X_i; \hat{f}))} \quad (19)$$

and weights $w(X_i) = p(X_i; \hat{f})(1 - p(X_i; \hat{f}))$, and carries out a weighted backfitting of (Z, X) with weights w . The weighted smooth is given by

$$\hat{P}_j = \frac{\mathcal{S}_j(wR_j)}{\mathcal{S}_j w}. \quad (20)$$

where \mathcal{S}_j is a linear smoothing matrix, such as a kernel smoother. This extends iteratively reweighted least squares to the nonparametric setting.

A sparsity penalty can be incorporated, just as for sparse additive models (SpAM) for regression. The Lagrangian is given by

$$\mathcal{L}(f, \lambda) = \mathbb{E} [\log(1 + e^{f(X)}) - Yf(X)] + \lambda \left(\sum_{j=1}^p \sqrt{\mathbb{E}(f_j^2(X_j))} - L \right) \quad (21)$$

and the stationary condition for component function f_j is $\mathbb{E}(p - Y | X_j) + \lambda v_j = 0$ where v_j is an element of the subgradient $\partial \sqrt{\mathbb{E}(f_j^2)}$. As in the unregularized case, this condition is

nonlinear in f , and so we linearize the gradient of the log-likelihood around \hat{f} . This yields the linearized condition $\mathbb{E}[w(X)(f(X) - Z) | X_j] + \lambda v_j = 0$. To see this, note that

$$0 = \mathbb{E} \left(p(X; \hat{f}) - Y + p(X; \hat{f})(1 - p(X; \hat{f}))(f(X) - \hat{f}(X)) | X_j \right) + \lambda v_j \quad (22)$$

$$= \mathbb{E}[w(X)(f(X) - Z) | X_j] + \lambda v_j \quad (23)$$

When $\mathbb{E}(f_j^2) \neq 0$, this implies the condition

$$\left(\mathbb{E}(w | X_j) + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}} \right) f_j(X_j) = \mathbb{E}(w R_j | X_j). \quad (24)$$

In the finite sample case, in terms of the smoothing matrix \mathcal{S}_j , this becomes

$$f_j = \frac{\mathcal{S}_j(w R_j)}{\mathcal{S}_j w + \lambda / \sqrt{\mathbb{E}(f_j^2)}}. \quad (25)$$

If $\|\mathcal{S}_j(w R_j)\| < \lambda$, then $f_j = 0$. Otherwise, this implicit, nonlinear equation for f_j cannot be solved explicitly, so one simply iterates until convergence:

$$f_j \leftarrow \frac{\mathcal{S}_j(w R_j)}{\mathcal{S}_j w + \lambda \sqrt{n} / \|f_j\|}. \quad (26)$$

When $\lambda = 0$, this yields the standard local scoring update (20).

Example 6 (SpAM for Spam) *Here we consider an email spam classification problem, using the logistic SpAM backfitting algorithm above. This dataset has been studied Hastie et al (2001) using a set of 3,065 emails as a training set, and conducting hypothesis tests to choose significant variables; there are a total of 4,601 observations with $p = 57$ attributes, all numeric. The attributes measure the percentage of specific words or characters in the email, the average and maximum run lengths of upper case letters, and the total number of such letters.*

The results of a typical run of logistic SpAM are summarized in Figure 8, using plug-in bandwidths. A held-out set is used to tune the regularization parameter λ .

11 Bagging and Random Forests

Suppose we draw B bootstrap samples and each time we construct a classifier. This gives classifiers h_1, \dots, h_B . We now classify by combining them:

$$h(x) = \begin{cases} 1 & \text{if } \frac{1}{B} \sum_j h_j(x) \geq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

$\lambda(\times 10^{-3})$	ERROR	# ZEROS	SELECTED VARIABLES
5.5	0.2009	55	{ 8,54}
5.0	0.1725	51	{ 8, 9, 27, 53, 54, 57}
4.5	0.1354	46	{7, 8, 9, 17, 18, 27, 53, 54, 57, 58}
4.0	0.1083 (\checkmark)	20	{4, 6–10, 14–22, 26, 27, 38, 53–58}
3.5	0.1117	0	ALL
3.0	0.1174	0	ALL
2.5	0.1251	0	ALL
2.0	0.1259	0	ALL

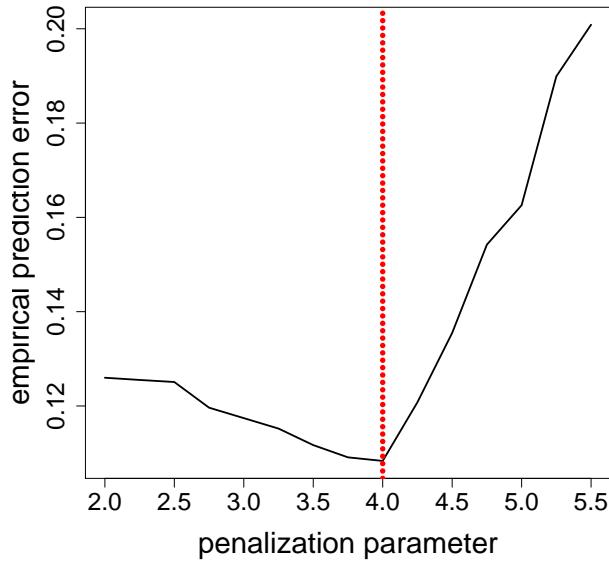


Figure 8: (Email spam) Classification accuracies and variable selection for logistic SpAM.

This is called *bagging* which stands for *bootstrap aggregation*. The baseline classifiers are usually trees.

A variation is to choose a random subset of the predictors to split on at each stage. The resulting classifier is called a random forests. Random forests often perform very well. Their theoretical performance is not well understood. Some good references are:

Biau, Devroye and Lugosi. (2008). Consistency of Random Forests and Other Average Classifiers. *JMLR*.

Biau, G. (2012). Analysis of a Random Forests Model. arXiv:1005.0208.

Lin and Jeon. Random Forests and Adaptive Nearest Neighbors. *Journal of the American Statistical Association*, 101, p 578.

Wager, S. (2014). Asymptotic Theory for Random Forests. arXiv:1405.0352.

Wager, S. (2015). Uniform convergence of random forests via adaptive concentration. arXiv:1503.06388.

Appendix: Multiclass Sparse Logistic Regression

Now we consider the multiclass version. Suppose we have the nonparametric K -class logistic regression model

$$p_f(Y = \ell | X) = \frac{e^{f_\ell(X)}}{\sum_{m=1}^K e^{f_m(X)}} \quad \ell = 1, \dots, K \quad (27)$$

where each function has an additive form

$$f_\ell(X) = f_{\ell 1}(X_1) + f_{\ell 2}(X_2) + \dots + f_{\ell p}(X_p). \quad (28)$$

In Newton's algorithm, we minimize the quadratic approximation to the log-likelihood

$$L(f) \approx L(\hat{f}) + \mathbb{E} \left[(Y - \hat{p})^T (f - \hat{f}) \right] + \frac{1}{2} \mathbb{E} \left[(f - \hat{f})^T H(\hat{f})(f - \hat{f}) \right] \quad (29)$$

where $\hat{p}(X) = (p_{\hat{f}}(Y = 1 | X), \dots, p_{\hat{f}}(Y = K | X))$, and $H(\hat{f}(X))$ is the Hessian

$$H(\hat{f}) = -\text{diag}(\hat{p}(X)) + \hat{p}(X)\hat{p}(X)^T. \quad (30)$$

Maximizing the right hand size of (29) is equivalent to minimizing

$$-\mathbb{E} \left((Y - \hat{p})^T (f - \hat{f}) \right) - \mathbb{E} \left(\hat{f}^T J f \right) + \frac{1}{2} \mathbb{E} \left(f^T J f \right) \quad (31)$$

which is, in turn, equivalent to minimizing the surrogate loss function

$$Q(f, \hat{f}) \equiv = \frac{1}{2} \mathbb{E} \left(\|Z - Af\|_2^2 \right). \quad (32)$$

where $J = -H(\hat{f})$, $A = J^{1/2}$, and Z is defined by

$$Z = J^{-1/2}(Y - \hat{p}) + J^{1/2}\hat{f} \quad (33)$$

$$= A^{-1}(Y - \hat{p}) + A\hat{f}. \quad (34)$$

The above calculation can be reexpressed as follows, which leads a multiclass backfitting algorithm. The difference in log-likelihoods for functions $\{\hat{f}_\ell\}$ and $\{f_\ell\}$ is, to second order,

$$\mathbb{E} \left[\sum_{\ell=0}^{K-1} p_\ell(X) \left(\hat{f}_\ell(X) - \sum_{k=0}^{K-1} p_k(X)\hat{f}_k(X) + \frac{Y_\ell - p_\ell(X)}{p_\ell(X)} - f_\ell(X) + \sum_{k=0}^{K-1} p_k(X)f_k(X) \right)^2 \right] \quad (35)$$

where $p_\ell(X) = \mathbb{P}(Y = \ell | X)$, and $Y_\ell = \delta(Y, \ell)$ are indicator variables. Minimizing over $\{f_\ell\}$ gives *coupled* equations for the functions f_ℓ ; they can't be solved independently over ℓ .

A practical approach is to use coordinate descent, computing the function f_ℓ holding the other functions $\{f_k\}_{k \neq \ell}$ fixed, and iterating. Assuming that $f_k = \hat{f}_k$ for $k \neq \ell$, this simplifies to

$$\mathbb{E} \left[p_\ell(1 - p_\ell)^2 \left(\hat{f}_\ell + \frac{Y_\ell - p_\ell}{p_\ell(1 - p_\ell)} - f_\ell \right)^2 + \sum_{k \neq \ell} p_k p_\ell^2 \left(\hat{f}_\ell + \frac{p_k - Y_k}{p_k p_\ell} - f_\ell \right)^2 \right]. \quad (36)$$

After some algebra, this can be seen to be the same as the usual objective function in the binary case, where we take $\hat{f}_0 = 1$ and \hat{f}_1 arbitrary.

Now assume f_ℓ (and \hat{f}_ℓ) has an additive form: $f_\ell(X) = \sum_{j=1}^p f_{\ell j}(X_j)$. Some further calculation shows that minimizing over each $f_{\ell j}$ yields the following backfitting algorithm:

$$f_{\ell j}(X_j) \leftarrow \frac{\mathbb{E} \left[p_\ell(1 - p_\ell) \left(\hat{f}_\ell - \sum_{k \neq j} f_{\ell k} + \frac{Y_\ell - p_\ell}{p_\ell(1 - p_\ell)} \right) | X_j \right]}{\mathbb{E} [p_\ell(1 - p_\ell) | X_j]}. \quad (37)$$

We approximate the conditional expectations by smoothing, as usual:

$$f_{\ell j}(x_j) \leftarrow \frac{\mathcal{S}_j(x_j)^T (w_\ell(X) R_{\ell j}(X))}{\mathcal{S}_j(x_j)^T (w_\ell(X))} \quad (38)$$

where

$$R_{\ell j}(X) = \hat{f}_\ell(X) - \sum_{k \neq j} f_{\ell k}(X_k) + \frac{Y_\ell - p_\ell(X)}{p_\ell(X)(1 - p_\ell(X))} \quad (39)$$

$$w_\ell(X) = p_\ell(X)(1 - p_\ell(X)). \quad (40)$$

This is the same as in binary logistic regression. We thus have the following algorithm:

MULTICLASS LOGISTIC BACKFITTING

1. Initialize $\{\hat{f}_\ell = 0\}$, and set $Z(X) = K$.

2. Iterate until convergence:

For each $\ell = 0, 1, \dots, K - 1$

A. Initialize $f_\ell = \hat{f}_\ell$

B. Iterate until convergence:

For each $j = 1, 2, \dots, p$

$$\begin{aligned} f_{\ell j}(x_j) &\leftarrow \frac{\mathcal{S}_j(x_j)^T (w_\ell(X) R_{\ell j}(X))}{\mathcal{S}_j(x_j)^T (w_\ell(X))} \text{ where} \\ R_{\ell j}(X) &= \hat{f}_\ell(X) - \sum_{k \neq j} f_{\ell k}(X_k) + \frac{Y_\ell - p_\ell(X)}{p_\ell(X)(1 - p_\ell(X))} \\ w_\ell(X) &= p_\ell(X)(1 - p_\ell(X)). \end{aligned}$$

C. Update $Z(X) \leftarrow Z(X) - e^{\hat{f}_\ell(X)} + e^{f_\ell(X)}$.

D. Set $\hat{f}_\ell \leftarrow f_\ell$.

Incrementally updating the normalizing constants (step C) is important so that the probabilities $p_\ell(X) = e^{\hat{f}_\ell(X)}/Z(X)$ can be efficiently computed, and we avoid an $O(K^2)$ algorithm. This can be extended to include a sparsity constraint, as in the binary case.

Random Forests

One of the best known classifiers is the *random forest*. It is very simple and effective but there is still a large gap between theory and practice. Basically, a random forest is an average of tree estimators.

These notes rely heavily on Biau and Scornet (2016) as well as the other references at the end of the notes.

1 Partitions and Trees

We begin by reviewing trees. As with nonparametric regression, simple and interpretable classifiers can be derived by partitioning the range of X . Let $\Pi_n = \{A_1, \dots, A_N\}$ be a partition of \mathcal{X} . Let A_j be the partition element that contains x . Then $\hat{h}(x) = 1$ if $\sum_{X_i \in A_j} Y_i \geq \sum_{X_i \in A_j} (1 - Y_i)$ and $\hat{h}(x) = 0$ otherwise. This is nothing other than the plugin classifier based on the partition regression estimator

$$\hat{m}(x) = \sum_{j=1}^N \bar{Y}_j I(x \in A_j)$$

where $\bar{Y}_j = n_j^{-1} \sum_{i=1}^n Y_i I(X_i \in A_j)$ is the average of the Y_i 's in A_j and $n_j = \#\{X_i \in A_j\}$. (We define \bar{Y}_j to be 0 if $n_j = 0$.)

Recall from the results on regression that if $m \in H_1(1, L)$ and the binwidth b of a regular partition satisfies $b \asymp n^{-1/(d+2)}$ then

$$\mathbb{E}\|\hat{m} - m\|_P^2 \leq \frac{c}{n^{2/(d+2)}}. \quad (1)$$

We conclude that the corresponding classification risk satisfies $R(\hat{h}) - R(h_*) = O(n^{-1/(d+2)})$.

Regression trees and classification trees (also called decision trees) are partition classifiers where the partition is built recursively. For illustration, suppose there are two covariates, $X_1 = \text{age}$ and $X_2 = \text{blood pressure}$. Figure 1 shows a classification tree using these variables.

The tree is used in the following way. If a subject has $\text{Age} \geq 50$ then we classify him as $Y = 1$. If a subject has $\text{Age} < 50$ then we check his blood pressure. If systolic blood pressure is < 100 then we classify him as $Y = 1$, otherwise we classify him as $Y = 0$. Figure 2 shows the same classifier as a partition of the covariate space.

Here is how a tree is constructed. First, suppose that there is only a single covariate X . We choose a split point t that divides the real line into two sets $A_1 = (-\infty, t]$ and $A_2 = (t, \infty)$. Let \bar{Y}_1 be the mean of the Y_i 's in A_1 and let \bar{Y}_2 be the mean of the Y_i 's in A_2 .

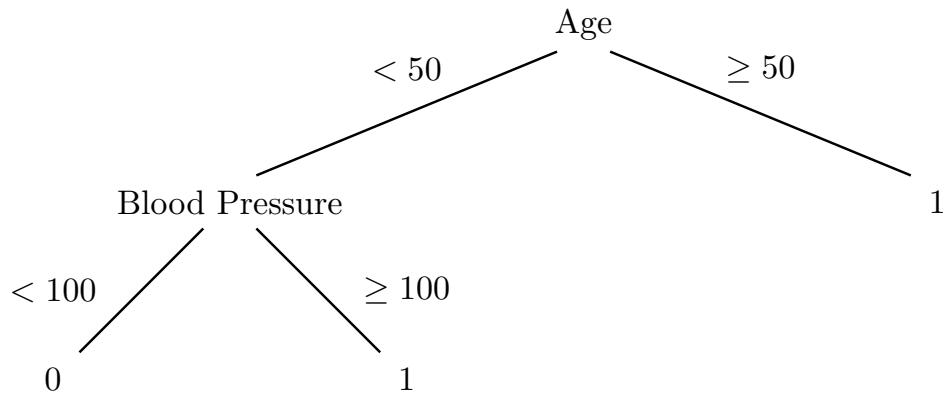


Figure 1: A simple classification tree.

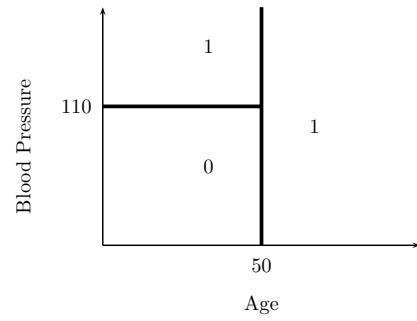


Figure 2: Partition representation of classification tree.

For continuous Y (regression), the split is chosen to minimize the training error. For binary Y (classification), the split is chosen to minimize a surrogate for classification error. A common choice is the impurity defined by $I(t) = \sum_{s=1}^2 \gamma_s$ where

$$\gamma_s = 1 - [\bar{Y}_s^2 + (1 - \bar{Y}_s)^2]. \quad (2)$$

This particular measure of impurity is known as the *Gini index*. If a partition element A_s contains all 0's or all 1's, then $\gamma_s = 0$. Otherwise, $\gamma_s > 0$. We choose the split point t to minimize the impurity. Other indices of impurity besides the Gini index can be used, such as entropy. The reason for using impurity rather than classification error is because impurity is a smooth function and hence is easy to minimize.

Now we continue recursively splitting until some stopping criterion is met. For example, we might stop when every partition element has fewer than n_0 data points, where n_0 is some fixed number. The bottom nodes of the tree are called the *leaves*. Each leaf has an estimate $\hat{m}(x)$ which is the mean of Y_i 's in that leaf. For classification, we take $\hat{h}(x) = I(\hat{m}(x) > 1/2)$. When there are several covariates, we choose whichever covariate and split that leads to the lowest impurity.

The result is a piecewise constant estimator that can be represented as a tree.

2 Example

The following data are from simulated images of gamma ray events for the Major Atmospheric Gamma-ray Imaging Cherenkov Telescope (MAGIC) in the Canary Islands. The data are from archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope. The telescope studies gamma ray bursts, active galactic nuclei and supernovae remnants. The goal is to predict if an event is real or is background (hadronic shower). There are 11 predictors that are numerical summaries of the images. We randomly selected 400 training points (200 positive and 200 negative) and 1000 test cases (500 positive and 500 negative). The results of various methods are in Table 1. See Figures 3, 4, 5, 6.

3 Bagging

Trees are useful for their simplicity and interpretability. But the prediction error can be reduced by combining many trees. A common approach, called bagging, is as follows.

Suppose we draw B bootstrap samples and each time we construct a classifier. This gives tree classifiers h_1, \dots, h_B . (The same idea applies to regression.) We now classify by combining

Method	Test Error
Logistic regression	0.23
SVM (Gaussian Kernel)	0.20
Kernel Regression	0.24
Additive Model	0.20
Reduced Additive Model	0.20
11-NN	0.25
Trees	0.20

Table 1: Various methods on the MAGIC data. The reduced additive model is based on using the three most significant variables from the additive model.

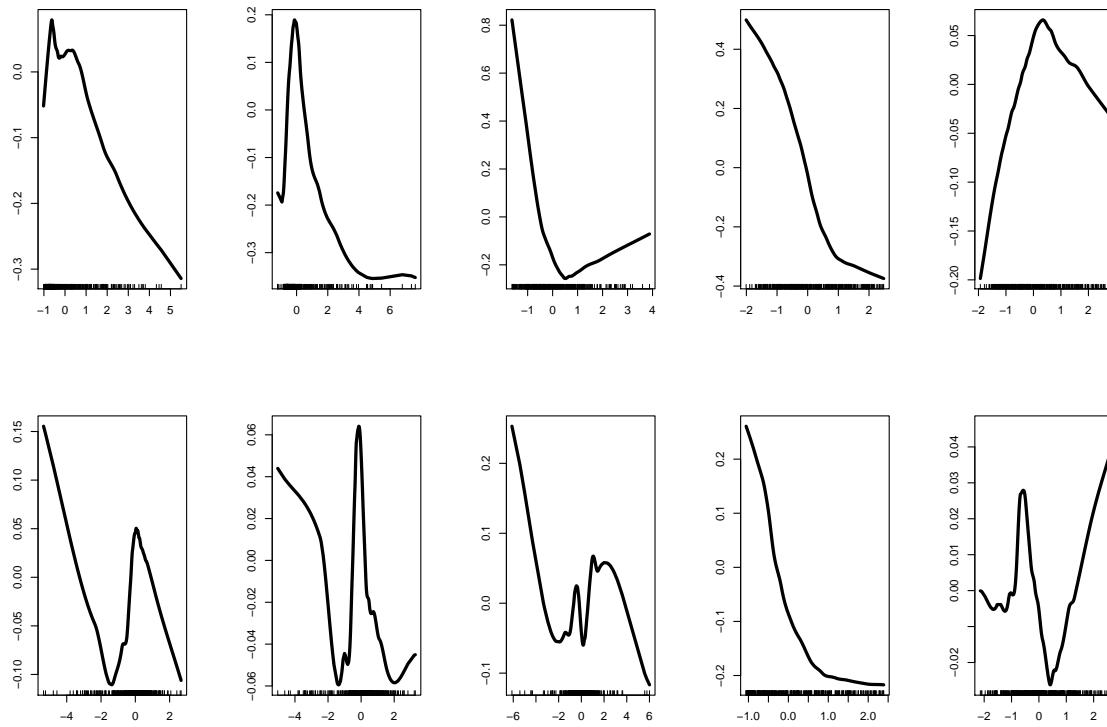


Figure 3: Estimated functions for additive model.

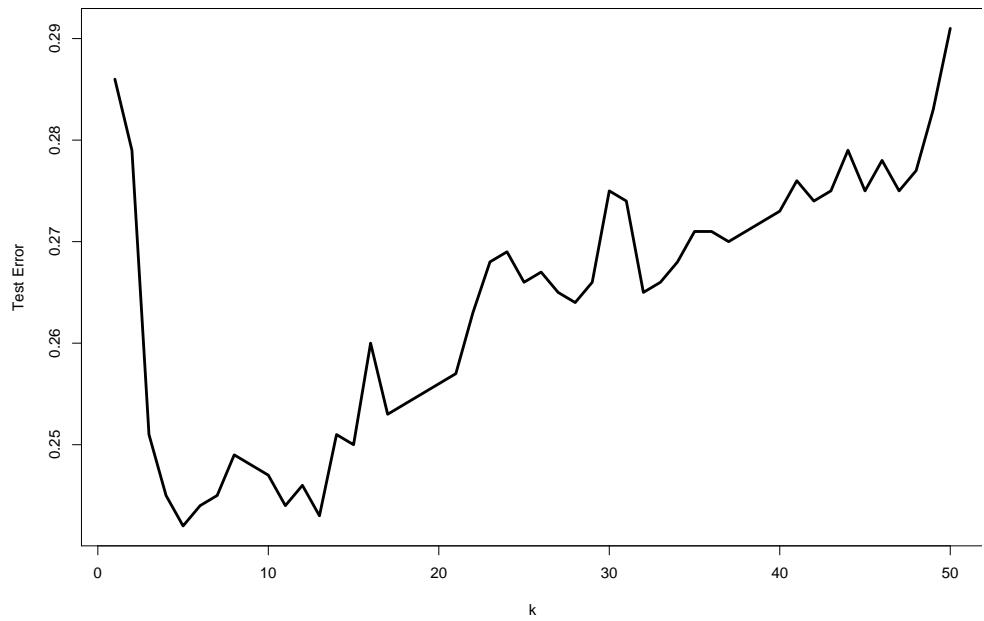


Figure 4: Test error versus k for nearest neighbor estimator.

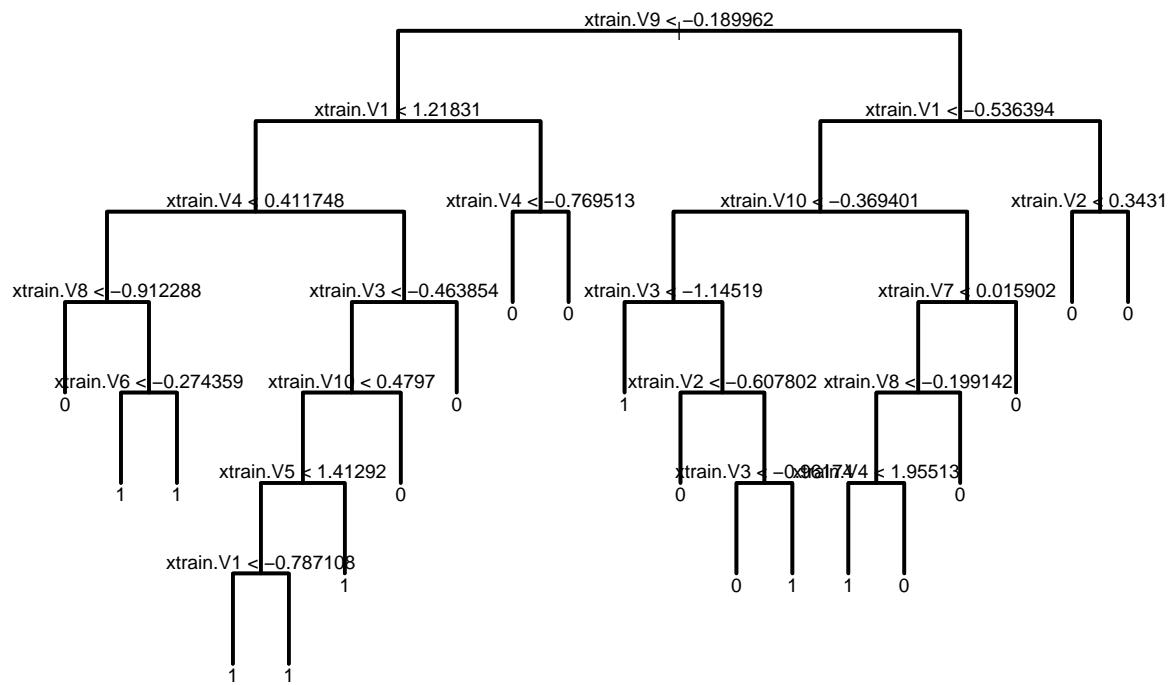


Figure 5: Full tree.

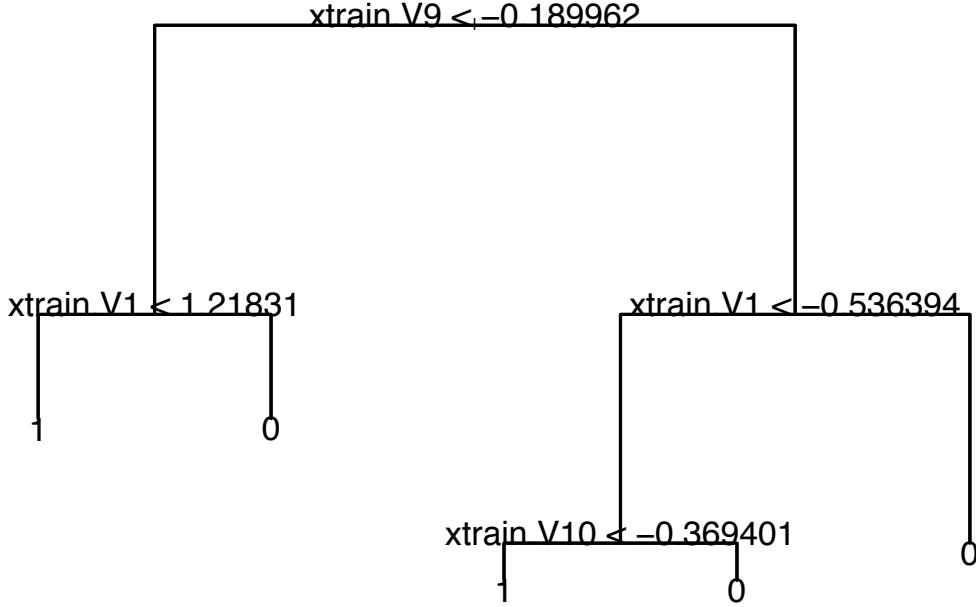


Figure 6: Classification tree. The size of the tree was chosen by cross-validation.

them:

$$h(x) = \begin{cases} 1 & \text{if } \frac{1}{B} \sum_j h_j(x) \geq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

This is called *bagging* which stands for *bootstrap aggregation*. A variation is sub-bagging where we use subsamples instead of bootstrap samples.

To get some intuition about why bagging is useful, consider this example from Buhlmann and Yu (2002). Suppose that $x \in \mathbb{R}$ and consider the simple decision rule $\hat{\theta}_n = I(\bar{Y}_n \leq x)$. Let $\mu = \mathbb{E}[Y_i]$ and for simplicity assume that $\text{Var}(Y_i) = 1$. Suppose that x is close to μ relative to the sample size. We can model this by setting $x \equiv x_n = \mu + c/\sqrt{n}$. Then $\hat{\theta}_n$ converges to $I(Z \leq c)$ where $Z \sim N(0, 1)$. So the limiting mean and variance of $\hat{\theta}_n$ are $\Phi(c)$ and $\Phi(c)(1 - \Phi(c))$. Now the bootstrap distribution of \bar{Y}^* (conditional on Y_1, \dots, Y_n) is approximately $N(\bar{Y}, 1/n)$. That is, $\sqrt{n}(\bar{Y}^* - \bar{Y}) \approx N(0, 1)$. Let E^* denote the average with respect to the bootstrap randomness. Then, if $\tilde{\theta}_n$ is the bagged estimator, we have

$$\begin{aligned} \tilde{\theta}_n &= E^*[I(\bar{Y}^* \leq x_n)] = E^*\left[I\left(\sqrt{n}(\bar{Y}^* - \bar{Y}) \leq \sqrt{n}(x_n - \bar{Y})\right)\right] \\ &= \Phi(\sqrt{n}(x_n - \bar{Y})) + o(1) = \Phi(c + Z) + o(1) \end{aligned}$$

where $Z \sim N(0, 1)$, and we used the fact that $\bar{Y} \approx N(\mu, 1/n)$.

To summarize, $\hat{\theta}_n \approx I(Z \leq c)$ while $\tilde{\theta}_n \approx \Phi(c + Z)$ which is a smoothed version of $I(Z \leq c)$.

In other words, bagging is a smoothing operator. In particular, suppose we take $c = 0$. Then $\widehat{\theta}_n$ converges to a Bernoulli with mean $1/2$ and variance $1/4$. The bagged estimator converges to $\Phi(Z) = \text{Unif}(0, 1)$ which has mean $1/2$ and variance $1/12$. The reduction in variance is due to the smoothing effect of bagging.

4 Random Forests

Finally we get to random forests. These are bagged trees except that we also choose random subsets of features for each tree. The estimator can be written as

$$\widehat{m}(x) = \frac{1}{M} \sum_j \widehat{m}_j(x)$$

where \widehat{m}_j is a tree estimator based on a subsample (or bootstrap) of size a using p randomly selected features. The trees are usually required to have some number k of observations in the leaves. There are three tuning parameters: a , p and k . You could also think of M as a tuning parameter but generally we can think of M as tending to ∞ .

For each tree, we can estimate the prediction error on the un-used data. (The tree is built on a subsample.) Averaging these prediction errors gives an estimate called the *out-of-bag* error estimate.

Unfortunately, it is very difficult to develop theory for random forests since the splitting is done using greedy methods. Much of the theoretical analysis is done using simplified versions of random forests. For example, the *centered forest* is defined as follows. Suppose the data are on $[0, 1]^d$. Choose a random feature, split in the center. Repeat until there are k leaves. This defines one tree. Now we average M such trees. Breiman (2004) and Biau (2002) proved the following.

Theorem 1 *If each feature is selected with probability $1/d$, $k = o(n)$ and $k \rightarrow \infty$ then*

$$\mathbb{E}[|\widehat{m}(X) - m(X)|^2] \rightarrow 0$$

as $n \rightarrow \infty$.

Under stronger assumptions we can say more:

Theorem 2 *Suppose that m is Lipschitz and that m only depends on a subset S of the features and that the probability of selecting $j \in S$ is $(1/S)(1 + o(1))$. Then*

$$\mathbb{E}|\widehat{m}(X) - m(X)|^2 = O\left(\frac{1}{n}\right)^{\frac{3}{4|S|\log 2+3}}.$$

This is better than the usual Lipschitz rate $n^{-2/(d+2)}$ if $|S| \leq p/2$. But the condition that we select relevant variables with high probability is very strong and proving that this holds is a research problem.

A significant step forward was made by Scornet, Biau and Vert (2015). Here is their result.

Theorem 3 Suppose that $Y = \sum_j m_j(X(j)) + \epsilon$ where $X \sim \text{Uniform}[0, 1]^d$, $\epsilon \sim N(0, \sigma^2)$ and each m_j is continuous. Assume that the split is chosen using the maximum drop in sums of squares. Let t_n be the number of leaves on each tree and let a_n be the subsample size. If $t_n \rightarrow \infty$, $a_n \rightarrow \infty$ and $t_n(\log a_n)^9/a_n \rightarrow 0$ then

$$\mathbb{E}[|\hat{m}(X) - m(X)|^2] \rightarrow 0$$

as $n \rightarrow \infty$.

Again, the theorem has strong assumptions but it does allow a greedy split selection. Scornet, Biau and Vert (2015) provide another interesting result. Suppose that (i) there is a subset S of relevant features, (ii) $p = d$, (iii) m_j is not constant on any interval for $j \in S$. Then with high probability, we always split only on relevant variables.

5 Connection to Nearest Neighbors

Lin and Jeon (2006) showed that there is a connection between random forests and k -NN methods. We say that X_i is a *layered nearest neighbor* (LNN) of x if the hyper-rectangle defined by x and X_i contains no data points except X_i . Now note that if tree is grown until each leaf has one point, then $\hat{m}(x)$ is simply a weighted average of LNN's. More generally, Lin and Jeon (2006) call X_i a k -potential nearest neighbor $k-PNN$ if there are fewer than k samples in the the hyper-rectangle defined by x and X_i . If we restrict to random forests whose leaves have k points then it follows easily that $\hat{m}(x)$ is some weighted average of the $k-PNN$'s.

Let us now return to LNN's. Let $\mathcal{L}_n(x)$ denote all LNN's of x and let $L_n(x) = |\mathcal{L}_n(x)|$. We could directly define

$$\hat{m}(x) = \frac{1}{L_n(x)} \sum_i Y_i I(X_i \in \mathcal{L}_n(x)).$$

Biau and Devroye (2010) showed that, if X has a continuous density,

$$\frac{(d-1)! \mathbb{E}[L_n(x)]}{2^d (\log n)^{d-1}} \rightarrow 1.$$

Moreover, if Y is bounded and m is continuous then, for all $p \geq 1$,

$$\mathbb{E}|\hat{m}_n(X) - m(X)|^p \rightarrow 0$$

as $n \rightarrow \infty$. Unfortunately, the rate of convergence is slow. Suppose that $\text{Var}(Y|X = x) = \sigma^2$ is constant. Then

$$\mathbb{E}|\hat{m}_n(X) - m(X)|^p \geq \frac{\sigma^2}{\mathbb{E}[L_n(x)]} \sim \frac{\sigma^2(d-1)!}{2^d(\log n)^{d-1}}.$$

If we use k -PNN, with $k \rightarrow \infty$ and $k = o(n)$, then the results Lin and Jeon (2006) show that the estimator is consistent and has variance of order $O(1/k(\log n)^{d-1})$.

As an aside, Biau and Devroye (2010) also show that if we apply bagging to the usual 1-NN rule to subsamples of size k and then average over subsamples, then, if $k \rightarrow \infty$ and $k = o(n)$, then for all $p \geq 1$ and all distributions P , we have that $\mathbb{E}|\hat{m}(X) - m(X)|^p \rightarrow 0$. So bagged 1-NN is universally consistent. But at this point, we have wondered quite far from random forests.

6 Connection to Kernel Methods

There is also a connection between random forests and kernel methods (Scornet 2016). Let $A_j(x)$ be the cell containing x in the j^{th} tree. Then we can write the tree estimator as

$$\hat{m}(x) = \frac{1}{M} \sum_j \sum_i \frac{Y_i I(X_i \in A_j(x))}{N_j(x)} = \frac{1}{M} \sum_j \sum_i W_{ij} Y_j$$

where $N_j(x)$ is the number of data points in $A_j(x)$ and $W_{ij} = I(X_i \in A_j(x))/N_j(x)$. This suggests that a cell A_j with low density (and hence small $N_j(x)$) has a high weight. Based on this observation, Scornet (2016) defined kernel based random forest (KeRF) by

$$\hat{m}(x) = \frac{\sum_j \sum_i Y_i I(X_i \in A_j(x))}{\sum_j N_j(x)}.$$

With this modification, $\hat{m}(x)$ is the average of each Y_i weighted by how often it appears in the trees. The KeRF can be written as

$$\hat{m}(x) = \frac{\sum_i Y_i K(x, X_i)}{\sum_s K_n(x, X_s)}$$

where

$$K_n(x, z) = \frac{1}{M} \sum_j I(x \in A_j(z)).$$

The trees are random. So let us write the j^{th} tree as $T_j = T(\Theta_j)$ for some random quantity Θ_j . So the forests is built from $T(\Theta_1), \dots, T(\Theta_M)$. And we can write $A_j(x)$ as $A(x, \Theta_j)$. Then $K_n(x, z)$ converges almost surely (as $M \rightarrow \infty$) to $\kappa_n(x, z) = P_\Theta(z \in A(x, \Theta))$ which is

just the probability that x and z are connected, in the sense that they are in the same cell. Under some assumptions, Scornet (2016) showed that KeRF's and forests are close to each other, thus providing a kernel interpretation of forests.

Recall the centered forest we discussed earlier. This is a stylized forest — quite different from the forests used in practice — but they provide a nice way to study the properties of the forest. In the case of KeRF's, Scornet (2016) shows that if $m(x)$ is Lipschitz and $X \sim \text{Unif}([0, 1]^d)$ then

$$\mathbb{E}[(\hat{m}(x) - m(x))^2] \leq C(\log n)^2 \left(\frac{1}{n}\right)^{\frac{1}{3+d \log 2}}.$$

This is slower than the minimax rate $n^{-2/(d+2)}$ but this probably reflects the difficulty in analyzing forests.

7 Variable Importance

Let \hat{m} be a random forest estimator. How important is feature $X(j)$?

LOCO. One way to answer this question is to fit the forest with all the data and fit it again without using $X(j)$. When we construct a forest, we randomly select features for each tree. This second forest can be obtained by simply average the trees where feature j was not selected. Call this $\hat{m}_{(-j)}$. Let \mathcal{H} be a hold-out sample of size m . Then let

$$\hat{\Delta}_j = \frac{1}{m} \sum_{i \in \mathcal{H}} W_i$$

where

$$W_i = (Y_i - \hat{m}_{(-j)}(X_i))^2 - (Y_i - \hat{m}(X_i))^2.$$

Then $\hat{\Delta}_j$ is a consistent estimate of the prediction risk inflation that occurs by not having access to $X(j)$. Formally, if \mathcal{T} denotes the training data then,

$$\mathbb{E}[\hat{\Delta}_j | \mathcal{T}] = \mathbb{E}\left[(Y - \hat{m}_{(-j)}(X))^2 - (Y - \hat{m}(X))^2 \mid \mathcal{T} \right] \equiv \Delta_j.$$

In fact, since $\hat{\Delta}_j$ is simply an average, we can easily construct a confidence interval. This approach is called LOCO (Leave-Out-COvariates). Of course, it is easily extended to sets of features. The method is explored in Lei, G'Sell, Rinaldo, Tibshirani, Wasserman (2017) and Rinaldo, Tibshirani, Wasserman (2015).

Permutation Importance. A different approach is to permute the values of $X(j)$ for the out-of-bag observations, for each tree. Let \mathcal{O}_j be the out-of-bag observations for tree j and

let \mathcal{O}_j^* be the out-of-bag observations for tree j with $X(j)$ permuted.

$$\widehat{\Gamma}_j = \frac{1}{M} \sum_j \sum_i W_{ij}$$

where

$$W_{ij} = \frac{1}{m_j} \sum_{i \in \mathcal{O}_j^*} (Y_i - \widehat{m}_j(X_i))^2 - \frac{1}{m_j} \sum_{i \in \mathcal{O}_j} (Y_i - \widehat{m}_j(X_i))^2.$$

This avoids using a hold-out sample. This is estimating

$$\Gamma_j = \mathbb{E}[(Y - \widehat{m}(X'_j))^2] - \mathbb{E}[(Y - \widehat{m}(X))^2]$$

where X'_j has the same distribution as X except that $X'_j(j)$ is an independent draw from $X(j)$. This is a lot like LOCO but its meaning is less clear. Note that \widehat{m}_j is not changed when $X(j)$ is permuted. Gregorutti, Michel and Saint Pierre. (2013) show that, when (X, ϵ) is Gaussian, that $\text{Var}(X) = (1 - c)I + c\mathbf{1}\mathbf{1}^T$ and that $\text{Cov}(Y, X(j)) = \tau$ for all j then

$$\Gamma_j = 2 \left(\frac{\tau}{1 - c + dc} \right)^2.$$

It is not clear how this connects to the actual importance of $X(j)$. In the case where $Y = \sum_j m_j(X(j)) + \epsilon$ with $\mathbb{E}[\epsilon|X] = 0$ and $\mathbb{E}[\epsilon^2|X] < \infty$, they show that $\Gamma_j = 2\text{Var}(m_j(X(j)))$.

8 Inference

Using the theory of infinite order U -statistics, Mentch and Hooker (2015) showed that $\sqrt{n}(\widehat{m}(x) - \mathbb{E}[\widehat{m}(x)])/\sigma$ converges to a Normal(0,1) and they show how to estimate σ .

Wager and Athey (2017) show asymptotic normality if we use sample splitting: part of the data are used to build the tree and part is used to estimate the average in the leafs of the tree. Under a number of technical conditions — including the fact that we use subsamples of size $s = n^\beta$ with $\beta < 1$ — they show that $(\widehat{m}(x) - m(x))/\sigma_n(x) \rightsquigarrow N(0, 1)$ and they show how to estimate $\sigma_n(x)$. Specifically,

$$\widehat{\sigma}_n^2(x) = \frac{n-1}{n} \left(\frac{n}{n-s} \right)^2 \sum_i (\text{Cov}(\widehat{m}_j(x), N_{ij})^2$$

where the covariance is with respect to the trees in the forest and $N_{ij} = 1$ if (X_i, Y_i) was in the j^{th} subsample and 0 otherwise.

9 Summary

Random forests are considered one of the best all purpose classifiers. But it is still a mystery why they work so well. The situation is very similar to deep learning. We have seen that there are now many interesting theoretical results about forests. But the results make strong assumptions that create a gap between practice and theory. Furthermore, there is no theory to say why forests outperform other methods. The gap between theory and practice is due to the fact that forests — as actually used in practice — are complex functions of the data.

10 References

- Biau, Devroye and Lugosi. (2008). Consistency of Random Forests and Other Average Classifiers. *JMLR*.
- Biau, Gerard, and Scornet. (2016). A random forest guided tour. *Test* 25.2: 197-227.
- Biau, G. (2012). Analysis of a Random Forests Model. arXiv:1005.0208.
- Buhlmann, P., and Yu, B. (2002). Analyzing bagging. *Annals of Statistics*, 927-961.
- Gregorutti, Michel, and Saint Pierre. (2013). Correlation and variable importance in random forests. arXiv:1310.5726.
- Lei J, G'Sell M, Rinaldo A, Tibshirani RJ, Wasserman L. (2017). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*.
- Lin, Y. and Jeon, Y. (2006). Random Forests and Adaptive Nearest Neighbors. *Journal of the American Statistical Association*, 101, p 578.
- L. Mentch and G. Hooker. (2015). Ensemble trees and CLTs: Statistical inference for supervised learning. *Journal of Machine Learning Research*.
- Rinaldo A, Tibshirani R, Wasserman L. (2015). Uniform asymptotic inference and the bootstrap after model selection. arXiv preprint arXiv:1506.06266.
- Scornet E. Random forests and kernel methods. (2016). *IEEE Transactions on Information Theory*. 62(3):1485-500.
- Wager, S. (2014). Asymptotic Theory for Random Forests. arXiv:1405.0352.
- Wager, S. (2015). Uniform convergence of random forests via adaptive concentration. arXiv:1503.06388.
- Wager, S. and Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*.

Clustering

10/26-702 Spring 2017

1 The Clustering Problem

In a clustering problem we aim to find groups in the data. Unlike classification, the data are not labeled, and so clustering is an example of *unsupervised learning*. We will study the following approaches:

1. k -means
2. Mixture models
3. Density-based Clustering I: Level Sets and Trees
4. Density-based Clustering II: Modes
5. Hierarchical Clustering
6. Spectral Clustering

Some issues that we will address are:

1. Rates of convergence
2. Choosing tuning parameters
3. Variable selection
4. High Dimensional Clustering

Example 1 Figures 17 and 18 show some synthetic examples where the clusters are meant to be intuitively clear. In Figure 17 there are two blob-like clusters. Identifying clusters like this is easy. Figure 18 shows four clusters: a blob, two rings and a half ring. Identifying clusters with unusual shapes like this is not quite as easy. In fact, finding clusters of this type requires nonparametric methods.

2 k-means (Vector Quantization)

One of the oldest approaches to clustering is to find k representative points, called *prototypes* or *cluster centers*, and then divide the data into groups based on which prototype they are closest to. For now, we assume that k is given. Later we discuss how to choose k .

Warning! My view is that k is a tuning parameter; it is **not** the number of clusters. Usually we want to choose k to be larger than the number of clusters.

Let $X_1, \dots, X_n \sim P$ where $X_i \in \mathbb{R}^d$. Let $C = \{c_1, \dots, c_k\}$ where each $c_j \in \mathbb{R}^d$. We call C a codebook. Let $\Pi_C[X]$ be the projection of X onto C :

$$\Pi_C[X] = \operatorname{argmin}_{c \in C} \|c - X\|^2. \quad (1)$$

Define the empirical clustering risk of a codebook C by

$$R_n(C) = \frac{1}{n} \sum_{i=1}^n \|X_i - \Pi_C[X_i]\|^2 = \frac{1}{n} \sum_{i=1}^n \min_{1 \leq j \leq k} \|X_i - c_j\|^2. \quad (2)$$

Let \mathcal{C}_k denote all codebooks of length k . The optimal codebook $\widehat{C} = \{\widehat{c}_1, \dots, \widehat{c}_k\} \in \mathcal{C}_k$ minimizes $R_n(C)$:

$$\widehat{C} = \operatorname{argmin}_{C \in \mathcal{C}_k} R_n(C). \quad (3)$$

The empirical risk is an estimate of the population clustering risk defined by

$$R(C) = \mathbb{E} \left\| X - \Pi_C[X] \right\|^2 = \mathbb{E} \min_{1 \leq j \leq k} \|X - c_j\|^2 \quad (4)$$

where $X \sim P$. The optimal population quantization $C^* = \{c_1^*, \dots, c_k^*\} \in \mathcal{C}_k$ minimizes $R(C)$. We can think of \widehat{C} as an estimate of C^* . This method is called k -means clustering or vector quantization.

A codebook $C = \{c_1, \dots, c_k\}$ defines a set of cells known as a *Voronoi tessellation*. Let

$$V_j = \left\{ x : \|x - c_j\| \leq \|x - c_s\|, \text{ for all } s \neq j \right\}. \quad (5)$$

The set V_j is known as a Voronoi cell and consists of all points closer to c_j than any other point in the codebook. See Figure 1.

The usual algorithm to minimize $R_n(C)$ and find \widehat{C} is the k -means clustering algorithm—also known as Lloyd’s algorithm—see Figure 2. The risk $R_n(C)$ has multiple minima. The algorithm will only find a local minimum and the solution depends on the starting values. A common way to choose the starting values is to select k data points at random. We will discuss better methods for choosing starting values in Section 2.1.

Example 2 Figure 3 shows synthetic data inspired by the Mickey Mouse example from http://en.wikipedia.org/wiki/K-means_clustering. The data in the top left plot form three clearly defined clusters. k -means easily finds in the clusters (top right). The bottom shows the same example except that we now make the groups very unbalanced. The lack of balance causes k -means to produce a poor clustering. But note that, if we “overfit then merge” then there is no problem.

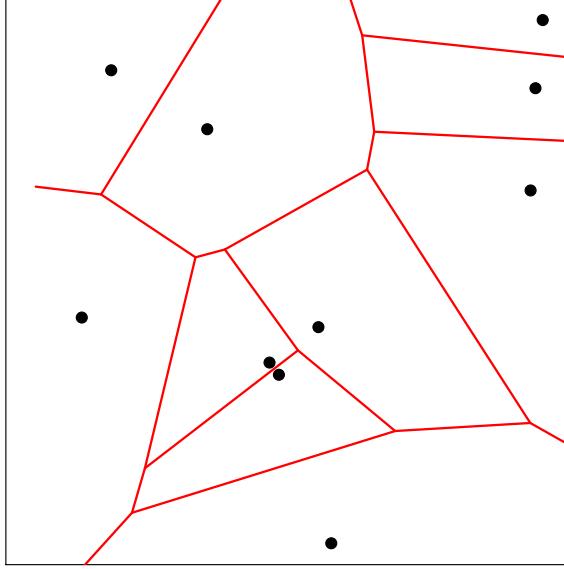


Figure 1: The Voronoi tessellation formed by 10 cluster centers c_1, \dots, c_{10} . The cluster centers are indicated by dots. The corresponding Voronoi cells T_1, \dots, T_{10} are defined as follows: a point x is in T_j if x is closer to c_j than c_i for $i \neq j$.

1. Choose k centers c_1, \dots, c_k as starting values.
2. Form the clusters C_1, \dots, C_k as follows. Let $g = (g_1, \dots, g_n)$ where $g_i = \operatorname{argmin}_j \|X_i - c_j\|$. Then $C_j = \{X_i : g_i = j\}$.
3. For $j = 1, \dots, k$, let n_j denote the number of points in C_j and set
$$c_j \leftarrow \frac{1}{n_j} \sum_{i: X_i \in C_j} X_i.$$
4. Repeat steps 2 and 3 until convergence.
5. Output: centers $\widehat{C} = \{c_1, \dots, c_k\}$ and clusters C_1, \dots, C_k .

Figure 2: The k -means (Lloyd's) clustering algorithm.

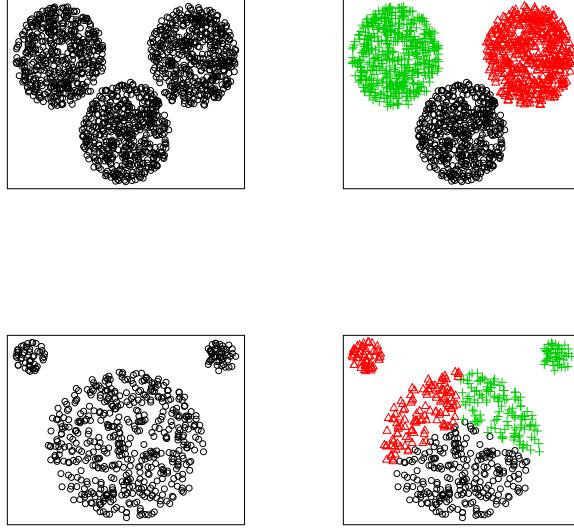


Figure 3: *Synthetic data inspired by the “Mickey Mouse” example from wikipedia.* Top left: three balanced clusters. Top right: result from running k means with $k = 3$. Bottom left: three unbalanced clusters. Bottom right: result from running k means with $k = 3$ on the unbalanced clusters. k -means does not work well here because the clusters are very unbalanced.

Example 3 We applied k -means clustering to the *Topex* data with $k = 9$. (*Topex* is a satellite.) The data are discretized so we treated each curve as one vector of length 70. The resulting nine clusters are shown in Figure 4.

Example 4 (Supernova Clustering) Figure 5 shows supernova data where we apply k -means clustering with $k = 4$. The type Ia supernovae get split into two groups although the groups are very similar. The other type also gets split into two groups which look qualitatively different.

Example 5 The top left plot of Figure 6 shows a dataset with two ring-shaped clusters. The remaining plots show the clusters obtained using k -means clustering with $k = 2, 3, 4$. Clearly, k -means does not capture the right structure in this case unless we overfit then merge.

2.1 Starting Values for k -means

Since $\widehat{R}_n(C)$ has multiple minima, Lloyd’s algorithm is not guaranteed to minimize $R_n(C)$. The clustering one obtains will depend on the starting values. The simplest way to choose starting values is to use k randomly chosen points. But this often leads to poor clustering.

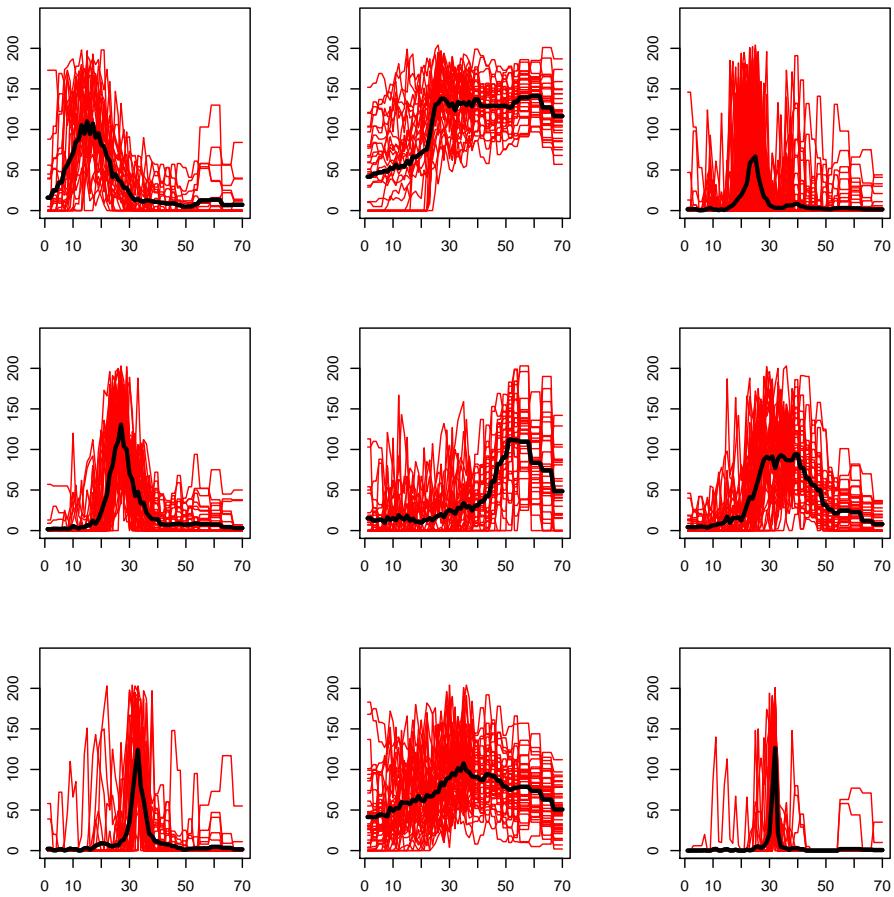


Figure 4: The nine clusters found in the Topex data using k -means clustering with $k = 9$. Each plot shows the curves in that cluster together with the mean of the curves in that cluster.

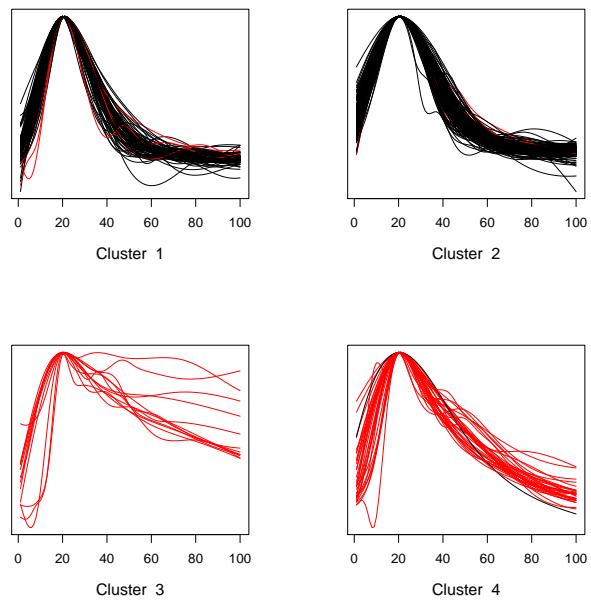


Figure 5: Clustering of the supernova light curves with $k = 4$.

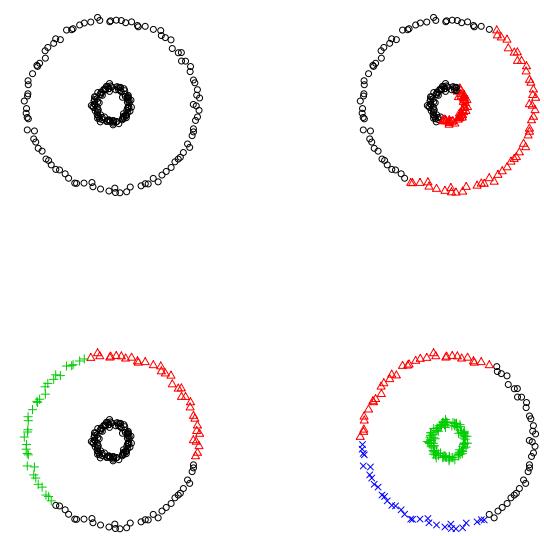


Figure 6: Top left: a dataset with two ring-shaped clusters. Top right: k -means with $k = 2$. Bottom left: k -means with $k = 3$. Bottom right: k -means with $k = 4$.

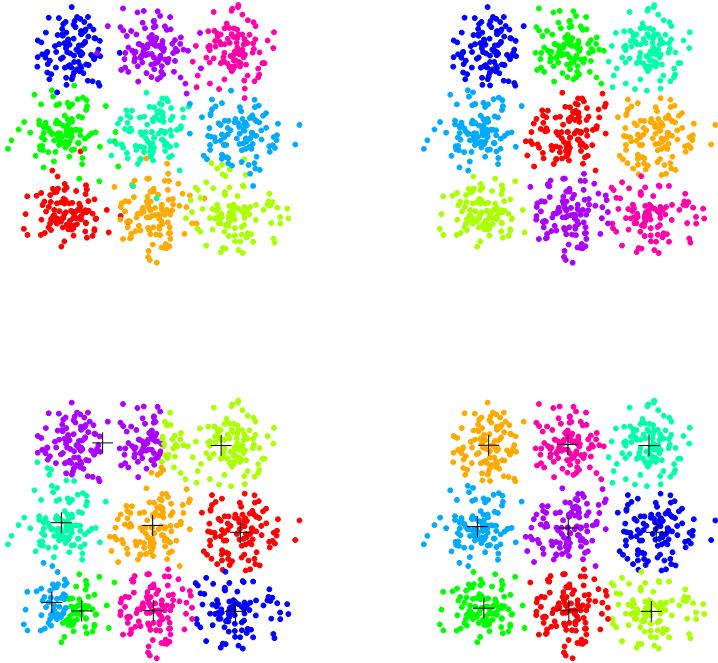


Figure 7: An example with 9 clusters. Top left: data. Top right: k -means with random starting values. Bottom left: k -means using starting values from hierarchical clustering. Bottom right: the k -means⁺⁺ algorithm.

Example 6 Figure 7 shows data from a distribution with nine clusters. The raw data are in the top left plot. The top right plot shows the results of running the k -means algorithm with $k = 9$ using random points as starting values. The clustering is quite poor. This is because we have not found the global minimum of the empirical risk function. The two bottom plots show better methods for selecting starting values that we will describe below.

Hierarchical Starting Values. Tseng and Wong (2005) suggest the following method for choosing starting values for k -means. Run single-linkage hierarchical clustering (which we describe in Section 6) to obtain $p \times k$ clusters. They suggest using $p = 3$ as a default. Now take the centers of the k -largest of the $p \times k$ clusters and use these as starting values. See the bottom left plot in Figure 7.

k -means⁺⁺. Arthur and Vassilvitskii (2007) invented an algorithm called k -means⁺⁺ to get good starting values. They show that if the starting points are chosen in a certain way, then we can get close to the minimum with high probability. In fact the starting points themselves — which we call seed points — are already close to minimizing $R_n(C)$. The algorithm is described in Figure 8. See the bottom right plot in Figure 7 for an example.

Theorem 7 (Arthur and Vassilvitskii, 2007). Let $C = \{c_1, \dots, c_k\}$ be the seed points from

1. Input: Data $X = \{X_1, \dots, X_n\}$ and an integer k .
2. Choose c_1 randomly from $X = \{X_1, \dots, X_n\}$. Let $C = \{c_1\}$.
3. For $j = 2, \dots, k$:
 - (a) Compute $D(X_i) = \min_{c \in C} \|X_i - c\|$ for each X_i .
 - (b) Choose a point X_i from X with probability

$$p_i = \frac{D^2(X_i)}{\sum_{j=1}^n D^2(X_j)}.$$
 - (c) Call this randomly chosen point c_j . Update $C \leftarrow C \cup \{c_j\}$.
4. Run Lloyd's algorithm using the **seed points** $C = \{c_1, \dots, c_k\}$ as starting points and output the result.

Figure 8: The k -means⁺⁺ algorithm.

the k -means⁺⁺ algorithm. Then,

$$\mathbb{E}(R_n(C)) \leq 8(\log k + 2) \left(\min_C R_n(C) \right) \quad (6)$$

where the expectation is over the randomness of the algorithm.

See Arthur and Vassilvitskii (2007) for a proof. They also show that the Euclidean distance can be replaced with the ℓ_p norm in the algorithm. The result is the same except that the constant 8 gets replaced by 2^{p+2} . It is possible to improve the k -means⁺⁺ algorithm. Ailon, Jaiswal and Monteleoni (2009) showed that, by choosing $3 \log k$ points instead of one point, at each step of the algorithm, the $\log k$ term in (6) can be replaced by a constant. They call the algorithm, k-means#.

2.2 Choosing k

In k -means clustering we must choose a value for k . This is still an active area of research and there are no definitive answers. The problem is much different than choosing a tuning parameter in regression or classification because there is no observable label to predict. Indeed, for k -means clustering, both the true risk R and estimated risk R_n decrease to 0

as k increases. This is in contrast to classification where the true risk gets large for high complexity classifiers even though the empirical risk decreases. Hence, minimizing risk does not make sense. There are so many proposals for choosing tuning parameters in clustering that we cannot possibly consider all of them here. Instead, we highlight a few methods.

Elbow Methods. One approach is to look for sharp drops in estimated risk. Let R_k denote the minimal risk among all possible clusterings and let \hat{R}_k be the empirical risk. It is easy to see that R_k is a nonincreasing function of k so minimizing R_k does not make sense. Instead, we can look for the first k such that the improvement $R_k - R_{k+1}$ is small, sometimes called an elbow. This can be done informally by looking at a plot of \hat{R}_k . We can try to make this more formal by fixing a small number $\alpha > 0$ and defining

$$k_\alpha = \min \left\{ k : \frac{R_k - R_{k+1}}{\sigma^2} \leq \alpha \right\} \quad (7)$$

where $\sigma^2 = \mathbb{E}(\|X - \mu\|^2)$ and $\mu = \mathbb{E}(X)$. An estimate of k_α is

$$\hat{k}_\alpha = \min \left\{ k : \frac{\hat{R}_k - \hat{R}_{k+1}}{\hat{\sigma}^2} \leq \alpha \right\} \quad (8)$$

where $\hat{\sigma}^2 = n^{-1} \sum_{i=1}^n \|X_i - \bar{X}\|^2$.

Unfortunately, the elbow method often does not work well in practice because there may not be a well-defined elbow.

Hypothesis Testing. A more formal way to choose k is by way of hypothesis testing. For each k we test

$$H_k : \text{the number of clusters is } k \quad \text{versus} \quad H_{k+1} : \text{the number of clusters is } > k.$$

We begin $k = 1$. If the test rejects, then we repeat the test for $k = 2$. We continue until the first k that is not rejected. In summary, \hat{k} is the first k for which k is not rejected.

Currently, my favorite approach is the one in Liu, Hayes, Andrew Nobel and Marron (2012). (JASA, 2102, 1281-1293). They simply test if the data are multivariate Normal. If this rejects, they split into two clusters and repeat. They have an R package `sigclust` for this. A similar procedure, called PG means is described in Feng and Hammerly (2007).

Example 8 *Figure 9 shows a two-dimensional example. The top left plot shows a single cluster. The p-values are shown as a function of k in the top right plot. The first k for which the p-value is larger than $\alpha = .05$ is $k = 1$. The bottom left plot shows a dataset with three clusters. The p-values are shown as a function of k in the bottom right plot. The first k for which the p-value is larger than $\alpha = .05$ is $k = 3$.*

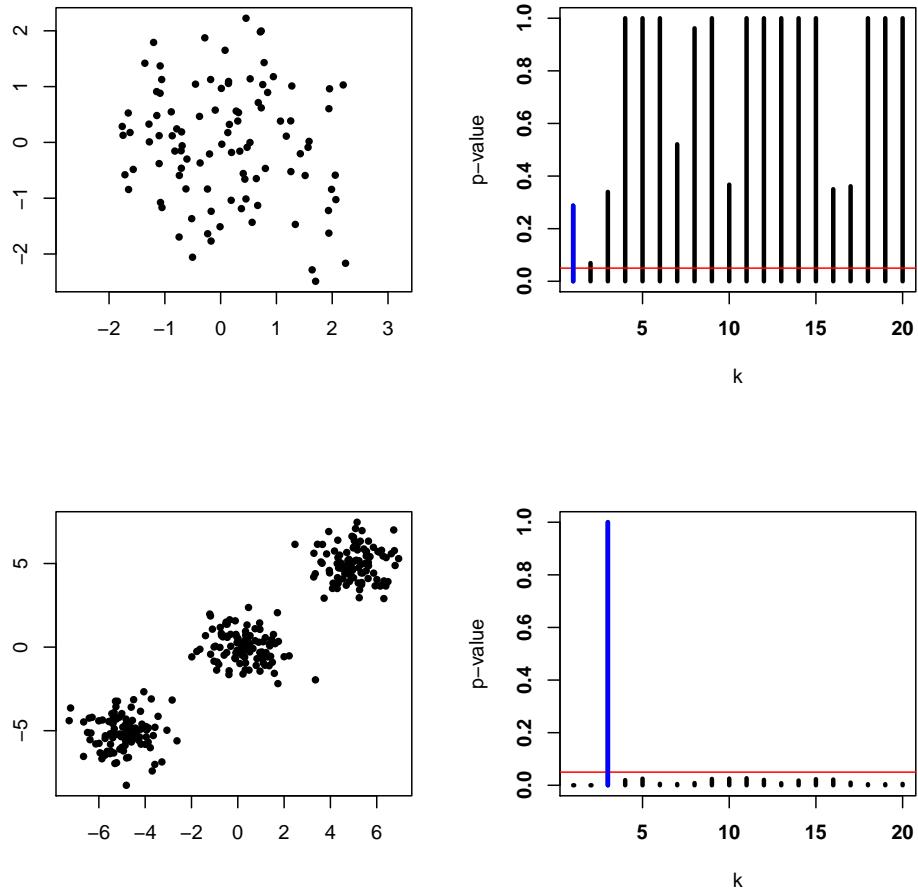


Figure 9: Top left: a single cluster. Top right: p-values for various k . The first k for which the p-value is larger than .05 is $k = 1$. Bottom left: three clusters. Bottom right: p-values for various k . The first k for which the p-value is larger than .05 is $k = 3$.

Stability. Another class of methods are based on the idea of stability. The idea is to find the largest number of clusters than can be estimated with low variability.

We start with a high level description of the idea and then we will discuss the details. Suppose that $Y = (Y_1, \dots, Y_n)$ and $Z = (Z_1, \dots, Z_n)$ are two independent samples from P . Let A_k be any clustering algorithm that takes the data as input and outputs k clusters. Define the *stability*

$$\Omega(k) = \mathbb{E}[s(A_k(Y), A_k(Z))] \quad (9)$$

where $s(\cdot, \cdot)$ is some measure of the similarity of two clusterings. To estimate Ω we use random subsampling. Suppose that the original data are $X = (X_1, \dots, X_{2n})$. Randomly split the data into two equal sets Y and Z of size n . This process if repeated N times. Denote the random split obtained in the j^{th} trial by Y^j, Z^j . Define

$$\widehat{\Omega}(k) = \frac{1}{N} \sum_{j=1}^N [s(A_k(Y^j), A_k(Z^j))].$$

For large N , $\widehat{\Omega}(k)$ will approximate $\Omega(k)$. There are two ways to choose k . We can choose a small k with high stability. Alternatively, we can choose k to maximize $\widehat{\Omega}(k)$ if we somehow standardize $\widehat{\Omega}(k)$.

Now we discuss the details. First, we need to define the similarity between two clusterings. We face two problems. The first is that the cluster labels are arbitrary: the clustering $(1, 1, 1, 2, 2, 2)$ is the same as the clustering $(4, 4, 4, 8, 8, 8)$. Second, the clusterings $A_k(Y)$ and $A_k(Z)$ refer to different data sets.

The first problem is easily solved. We can insist the labels take values in $\{1, \dots, k\}$ and then we can maximize the similarity over all permutations of the labels. Another way to solve the problem is the following. Any clustering method can be regarded as a function ψ that takes two points x and y and outputs a 0 or a 1. The interpretation is that $\psi(x, y) = 1$ if x and y are in the same cluster while $\psi(x, y) = 0$ if x and y are in a different cluster. Using this representation of the clustering renders the particular choice of labels moot. This is the approach we will take.

Let ψ_Y and ψ_Z be clusterings derived from Y and Z . Let us think of Y as training data and Z as test data. Now ψ_Y returns a clustering for Y and ψ_Z returns a clustering for Z . We'd like to somehow apply ψ_Y to Z . Then we would have two clusterings for Z which we could then compare. There is no unique way to do this. A simple and fairly general approach is to define

$$\psi_{Y,Z}(Z_j, Z_k) = \psi_Y(Y'_j, Y'_k) \quad (10)$$

where Y'_j is the closest point in Y to Z_j and Y'_k is the closest point in Y to Z_k . (More generally, we can use Y and the cluster assignment to Y as input to a classifier; see Lange et al 2004). The notation $\psi_{Y,Z}$ indicates that ψ is trained on Y but returns a clustering for

Z . Define

$$s(\psi_{Y,Z}, \psi_Z) = \frac{1}{\binom{n}{2}} \sum_{s \neq t} I(\psi_{Y,Z}(Z_s, Z_t) = \psi_Z(Z_s, Z_t)).$$

Thus s is the fraction of pairs of points in Z on which the two clusterings $\psi_{Y,Z}$ and ψ_Z agree. Finally, we define

$$\widehat{\Omega}(k) = \frac{1}{N} \sum_{j=1}^N s(\psi_{Y^j, Z^j}, \psi_{Z^j}).$$

Now we need to decide how to use $\widehat{\Omega}(k)$ to choose k . The interpretation of $\widehat{\Omega}(k)$ requires some care. First, note that $0 \leq \widehat{\Omega}(k) \leq 1$ and $\widehat{\Omega}(1) = \widehat{\Omega}(n) = 1$. So simply maximizing $\widehat{\Omega}(k)$ does not make sense. One possibility is to look for a small k larger than $k > 1$ with a high stability. Alternatively, we could try to normalize $\widehat{\Omega}(k)$. Lange et al (2004) suggest dividing by the value of $\widehat{\Omega}(k)$ obtained when cluster labels are assigned randomly. The theoretical justification for this choice is not clear. Tibshirani, Walther, Botstein and Brown (2001) suggest that we should compute the stability separately over each cluster and then take the minimum. However, this can sometimes lead to very low stability for all $k > 1$.

Many authors have considered schemes of this form, including Breckenridge (1989), Lange, Roth, Braun and Buhmann (2004), Ben-Hur, Elisseeff and Guyron (2002), Dudoit and Fridlyand (2002), Levine and Domany (2001), Buhmann (2010), Tibshirani, Walther, Botstein and Brown (2001) and Rinaldo and Wasserman (2009).

It is important to interpret stability correctly. These methods choose the largest number of stable clusters. That does not mean they choose “the true k .” Indeed, Ben-David, von Luxburg and Pál (2006), Ben-David and von Luxburg Tübingen (2008) and Raklin (2007) have shown that trying to use stability to choose “the true k ” — even if that is well-defined — will not work. To explain this point further, we consider some examples from Ben-David, von Luxburg and Pál (2006). Figure 10 shows the four examples. The first example (top left plot) shows a case where we fit $k = 2$ clusters. Here, stability analysis will correctly show that k is too small. The top right plot has $k = 3$. Stability analysis will correctly show that k is too large. The bottom two plots show potential failures of stability analysis. Both cases are stable but $k = 2$ is too small in the bottom left plot and $k = 3$ is too big in the bottom right plot. Stability is subtle. There is much potential for this approach but more work needs to be done.

2.3 Theoretical Properties

A theoretical property of the k -means method is given in the following result. Recall that $C^* = \{c_1^*, \dots, c_k^*\}$ minimizes $R(C) = \mathbb{E}\|X - \Pi_C[X]\|^2$.

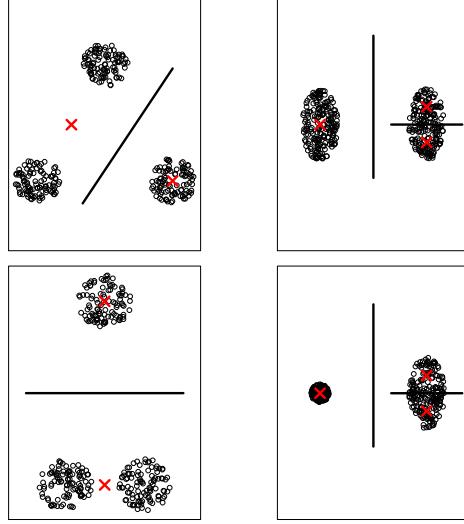


Figure 10: Examples from Ben-David, von Luxburg and Pál (2006). The first example (top left plot) shows a case where we fit $k = 2$ clusters. Stability analysis will correctly show that k is too small. The top right plot has $k = 3$. Stability analysis will correctly show that k is too large. The bottom two plots show potential failures of stability analysis. Both cases are stable but $k = 2$ is too small in the bottom left plot and $k = 3$ is too big in the bottom right plot.

Theorem 9 Suppose that $\mathbb{P}(\|X_i\|^2 \leq B) = 1$ for some $B < \infty$. Then

$$\mathbb{E}(R(\hat{C})) - R(C^*) \leq c \sqrt{\frac{k(d+1) \log n}{n}} \quad (11)$$

for some $c > 0$.

Warning! The fact that $R(\hat{C})$ is close to $R(C_*)$ does not imply that \hat{C} is close to C_* .

This proof is due to Linder, Lugosi and Zeger (1994). The proof uses techniques from a later lecture on VC theory so you may want to return to the proof later.

Proof. Note that $R(\hat{C}) - R(C^*) = R(\hat{C}) - R_n(\hat{C}) + R_n(\hat{C}) - R(C^*) \leq R(\hat{C}) - R_n(\hat{C}) + R_n(C^*) - R(C^*) \leq 2 \sup_{C \in \mathcal{C}_k} |R(\hat{C}) - R_n(\hat{C})|$. For each C define a function f_C by $f_C(x) = \|x - \Pi_C[x]\|^2$. Note that $\sup_x |f_C(x)| \leq 4B$ for all C . Now, using the fact that $\mathbb{E}(Y) =$

$\int_0^\infty \mathbb{P}(Y \geq t) dt$ whenever $Y \geq 0$, we have

$$\begin{aligned}
2 \sup_{C \in \mathcal{C}_k} |R(\widehat{C}) - R_n(\widehat{C})| &= 2 \sup_C \left| \frac{1}{n} \sum_{i=1}^n f_C(X_i) - \mathbb{E}(f_C(X)) \right| \\
&= 2 \sup_C \left| \int_0^\infty \left(\frac{1}{n} \sum_{i=1}^n I(f_C(X_i) > u) - \mathbb{P}(f_C(Z) > u) \right) du \right| \\
&\leq 8B \sup_{C,u} \left| \frac{1}{n} \sum_{i=1}^n I(f_C(X_i) > u) - \mathbb{P}(f_C(Z) > u) \right| \\
&= 8B \sup_A \left| \frac{1}{n} \sum_{i=1}^n I(X_i \in A) - \mathbb{P}(A) \right|
\end{aligned}$$

where A varies over all sets \mathcal{A} of the form $\{f_C(x) > u\}$. The shattering number of \mathcal{A} is $s(\mathcal{A}, n) \leq n^{k(d+1)}$. This follows since each set $\{f_C(x) > u\}$ is a union of the complements of k spheres. By the VC Theorem,

$$\begin{aligned}
\mathbb{P}(R(\widehat{C}) - R(C^*) > \epsilon) &\leq \mathbb{P} \left(8B \sup_A \left| \frac{1}{n} \sum_{i=1}^n I(X_i \in A) - \mathbb{P}(A) \right| > \epsilon \right) \\
&= \mathbb{P} \left(\sup_A \left| \frac{1}{n} \sum_{i=1}^n I(X_i \in A) - \mathbb{P}(A) \right| > \frac{\epsilon}{8B} \right) \\
&\leq 4(2n)^{k(d+1)} e^{-n\epsilon^2/(512B^2)}.
\end{aligned}$$

Now conclude that $\mathbb{E}(R(\widehat{C}) - R(C^*)) \leq C \sqrt{k(d+1)} \sqrt{\frac{\log n}{n}}$. \square

A sharper result, together with a lower bound is the following.

Theorem 10 (Bartlett, Linder and Lugosi 1997) *Suppose that $\mathbb{P}(\|X\|^2 \leq 1) = 1$ and that $n \geq k^{4/d}$, $\sqrt{dk^{1-2/d} \log n} \geq 15$, $kd \geq 8$, $n \geq 8d$ and $n/\log n \geq dk^{1+2/d}$. Then,*

$$\mathbb{E}(R(\widehat{C})) - R(C^*) \leq 32 \sqrt{\frac{dk^{1-2/d} \log n}{n}} = O\left(\sqrt{\frac{dk \log n}{n}}\right).$$

Also, if $k \geq 3$, $n \geq 16k/(2\Phi^2(-2))$ then, for any method \widehat{C} that selects k centers, there exists P such that

$$\mathbb{E}(R(\widehat{C})) - R(C^*) \geq c_0 \sqrt{\frac{k^{1-4/d}}{n}}$$

where $c_0 = \Phi^4(-2)2^{-12}/\sqrt{6}$ and Φ is the standard Gaussian distribution function.

See Bartlett, Linder and Lugosi (1997) for a proof. It follows that k -means is risk consistent in the sense that $R(\widehat{C}) - R(C^*) \xrightarrow{P} 0$, as long as $k = o(n/(d^3 \log n))$. Moreover, the lower

bound implies that we cannot find any other method that improves much over the k -means approach, at least with respect to this loss function.

The k -means algorithm can be generalized in many ways. For example, if we replace the L_2 norm with the L_1 norm we get k -medians clustering. We will not discuss these extensions here.

2.4 Overfitting and Merging

The best way to use k -means clustering is to “overfit then merge.” Don’t think of the k in k -means as the number of clusters. Think of it as a tuning parameter. k -means clustering works much better if we:

1. Choose k large
2. merge close clusters

This eliminates the sensitivity to the choice of k and it allows k -means to fit clusters with arbitrary shapes. Currently, there is no definitive theory for this approach but in my view, it is the right way to do k -means clustering.

3 Mixture Models

Simple cluster structure can be discovered using mixture models. We start with a simple example. We flip a coin with success probability π . If heads, we draw X from a density $p_1(x)$. If tails, we draw X from a density $p_0(x)$. Then the density of X is

$$p(x) = \pi p_1(x) + (1 - \pi)p_0(x),$$

which is called a mixture of two densities p_1 and p_0 . Figure 11 shows a mixture of two Gaussians distribution.

Let $Z \sim \text{Bernoulli}(\pi)$ be the unobserved coin flip. Then we can also write $p(x)$ as

$$p(x) = \sum_{z=0,1} p(x, z) = \sum_{z=0,1} p(x|z)p(z) \tag{12}$$

where $p(x|Z = 0) := p_0(x)$, $p(x|Z = 1) := p_1(x)$ and $p(z) = \pi^z(1 - \pi)^{1-z}$. Equation (12) is called the hidden variable representation. A more formal definition of finite mixture models is as follows.

[Finite Mixture Models] Let $\{p_\theta(x) : \theta \in \Theta\}$ be a parametric class of densities. Define the mixture model

$$p_\psi(x) = \sum_{j=0}^{K-1} \pi_j p_{\theta_j}(x),$$

where the mixing coefficients $\pi_j \geq 0$, $\sum_{j=0}^{K-1} \pi_j = 1$ and $\psi = (\pi_0, \dots, \pi_{K-1}, \theta_0, \dots, \theta_{K-1})$ are the unknown parameters. We call $p_{\theta_0}, \dots, p_{\theta_{K-1}}$ the component densities.

Generally, even if $\{p_\theta(x) : \theta \in \Theta\}$ is an exponential family model, the mixture may no longer be an exponential family.

3.1 Mixture of Gaussians

Let $\phi(x; \mu_j, \sigma_j^2)$ be the probability density function of a univariate Gaussian distribution with mean μ_j and variance σ_j^2 . A typical finite mixture model is the mixture of Gaussians. In one dimension, we have

$$p_\psi(x) = \sum_{j=0}^{K-1} \pi_j \phi(x; \mu_j, \sigma_j^2),$$

which has $3K - 1$ unknown parameters, due to the restriction $\sum_{j=0}^{K-1} \pi_j = 1$.

A mixture of d -dimensional multivariate Gaussians is

$$p(x) = \sum_{j=0}^{K-1} \frac{\pi_j}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \exp \left\{ -\frac{1}{2} (x - u_j)^T \Sigma_j^{-1} (x - u_j) \right\}.$$

There are in total

$$K \left(\underbrace{\frac{d(d+1)}{2}}_{\# \text{ of parameters in } \Sigma_j} + \underbrace{d}_{\# \text{ of parameters in } u_j} \right) + \underbrace{(K-1)}_{\# \text{ of mixing coefficients}} = \frac{Kd(d+3)}{2} + K - 1$$

parameters in the mixture of K multivariate Gaussians.

3.2 Maximum Likelihood Estimation

A finite mixture model $p_\psi(x)$ has parameters $\psi = (\pi_0, \dots, \pi_{K-1}, \theta_0, \dots, \theta_{K-1})$. The likelihood of ψ based on the observations X_1, \dots, X_n is

$$\mathcal{L}(\psi) = \prod_{i=1}^n p_\psi(X_i) = \prod_{i=1}^n \left(\sum_{j=0}^{K-1} \pi_j p_{\theta_j}(X_i) \right)$$

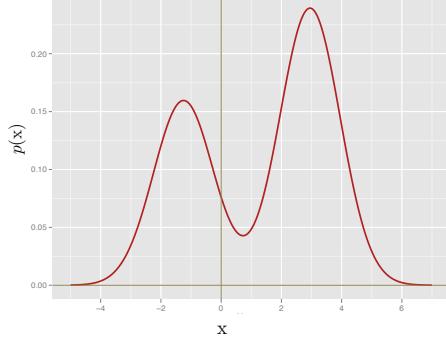


Figure 11: A mixture of two Gaussians, $p(x) = \frac{2}{5}\phi(x; -1.25, 1) + \frac{3}{5}\phi(x; 2.95, 1)$.

and, as usual, the maximum likelihood estimator is the value $\hat{\psi}$ that maximizes $\mathcal{L}(\psi)$. Usually, the likelihood is multimodal and one seeks a local maximum instead if a global maximum.

For fixed $\theta_0, \dots, \theta_{K-1}$, the log-likelihood is often a concave function of the mixing parameters π_j . However, for fixed π_0, \dots, π_{K-1} , the log-likelihood is not generally concave with respect to $\theta_0, \dots, \theta_{K-1}$.

One way to find $\hat{\psi}$ is to apply your favorite optimizer directly to the log-likelihood.

$$\ell(\psi) = \sum_{i=1}^n \log \left(\sum_{j=0}^{K-1} \pi_j p_{\theta_j}(X_i) \right).$$

However, $\ell(\psi)$ is not jointly convex with respect to ψ . It is not clear which algorithm is the best to optimize such a nonconvex objective function.

A convenient and commonly used algorithm for finding the maximum likelihood estimates of a mixture model (or the more general latent variable models) is the *expectation-maximization (EM)* algorithm. The algorithm runs in an iterative fashion and alternates between the “E-step” which computes conditional expectations with respect to the current parameter estimate, and the “M-step” which adjusts the parameter to maximize a lower bound on the likelihood. While the algorithm can be slow to converge, its simplicity and the fact that it doesn’t require a choice of step size make it a convenient choice for many estimation problems.

On the other hand, while simple and flexible, the EM algorithm is only one of many numerical procedures for obtaining a (local) maximum likelihood estimate of the latent variable models. In some cases procedures such as Newton’s method or conjugate gradient may be more effective, and should be considered as alternatives to EM. In general the EM algorithm converges linearly, and may be extremely slow when the amount of missing information is large,

In principle, there are polynomial time algorithms for finding good estimates of ψ based on spectral methods and the method of moments. It appears that, at least so far, these methods

are not yet practical enough to be used in routine data analysis.

Example. The data are measurements on duration and waiting time of eruptions of the Old Faithful geyser from August 1 to August 15, 1985. There are two variables with 299 observations. The first variable , “Duration”, represents the numeric eruption time in minutes. The second variable, “waiting”, represents the waiting time to next eruption. This data is believed to have two modes. We fit a mixture of two Gaussians using EM algorithm. To illustrate the EM step, we purposely choose a bad starting point. The EM algorithm quickly converges in six steps. Figure 12 illustrates the fitted densities for all the six steps. We see that even though the starting density is unimodal, it quickly becomes bimodal.

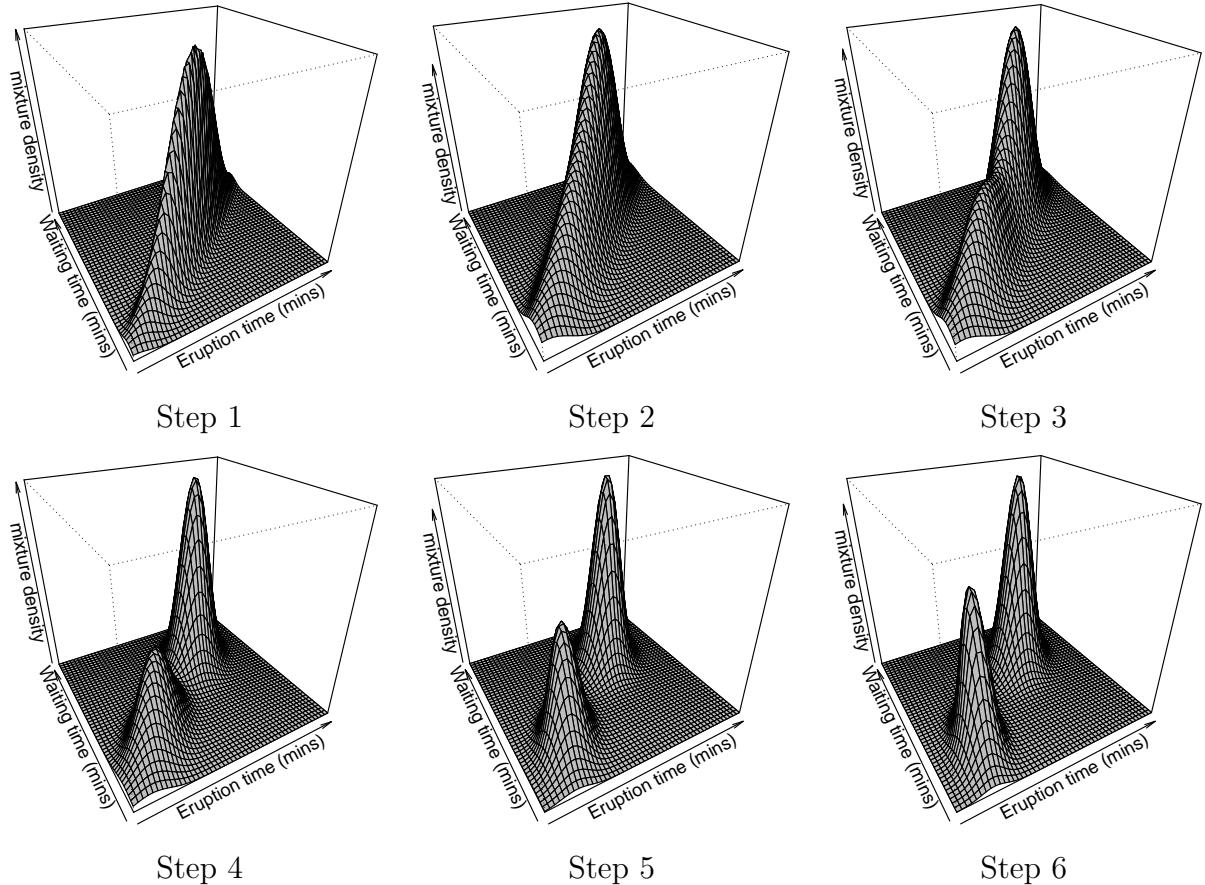


Figure 12: Fitting a mixture of two Gaussians on the Old Faithful Geyser data. The initial values are $\pi_0 = \pi_1 = 0.5$. $u_0 = (4, 70)^T$, $u_1 = (3, 60)^T$, $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 0.8 & 7 \\ 7 & 70 \end{pmatrix}$. We see that even though the starting density is not bimodal, the EM algorithm converges quickly to a bimodal density.

3.3 The Twilight Zone

Mixtures models are conceptually simple but they have some strange properties.

Computation. Finding the mle is NP-hard.

Infinite Likelihood. Let $p_\psi(x) = \sum_{j=1}^k \pi_j \phi(x; \mu_j, \sigma_j^2)$, be a mixture of Gaussians. Let $\mathcal{L}(\psi) = \prod_{i=1}^n p_\psi(X_i)$ be the likelihood function based on a sample of size n . Then $\sup_\psi \mathcal{L}(\psi) = \infty$. To see this, set $\mu_j = X_1$ for some j . Then $\phi(X_1; \mu_j, \sigma_j^2) = (\sqrt{2\pi}\sigma_j)^{-1}$. Now let $\sigma_j \rightarrow 0$. We have $\phi(X_1; \mu_j, \sigma_j^2) \rightarrow \infty$. Therefore, the log-likelihood is unbounded. This behavior is very different from a typical parametric model. Fortunately, if we define the maximum likelihood estimate to be a mode of $\mathcal{L}(\psi)$ in the interior of the parameter space, we get a well-defined estimator.

Multimodality of the Density. Consider the mixture of two Gaussians

$$p(x) = (1 - \pi)\phi(x; \mu_1, \sigma^2) + \pi\phi(x; \mu_0, \sigma^2).$$

You would expect $p(x)$ to be multimodal but this is not necessarily true. The density $p(x)$ is unimodal when $|\mu_1 - \mu_2| \leq 2\sigma$ and bimodal when $|\mu_1 - \mu_2| > 2\sigma$. One might expect that the maximum number of modes of a mixture of k Gaussians would be k . However, there are examples where a mixture of k Gaussians has more than k modes. In fact, Edelsbrunner, Fasy and Rote (2012) show that the relationship between the number of modes of p and the number of components in the mixture is very complex.

Nonidentifiability. A model $\{p_\theta(x) : \theta \in \Theta\}$ is identifiable if

$$\theta_1 \neq \theta_2 \text{ implies } P_{\theta_1} \neq P_{\theta_2}$$

where P_θ is the distribution corresponding to the density p_θ . Mixture models are nonidentifiable in two different ways. First, there is nonidentifiability due to permutation of labels. For example, consider a mixture of two univariate Gaussians,

$$p_{\psi_1}(x) = 0.3\phi(x; 0, 1) + 0.7\phi(x; 2, 1)$$

and

$$p_{\psi_2}(x) = 0.7\phi(x; 2, 1) + 0.3\phi(x; 0, 1),$$

then $p_{\psi_1}(x) = p_{\psi_2}(x)$ even though $\psi_1 = (0.3, 0.7, 0, 2, 1)^T \neq (0.7, 0.3, 2, 0, 1)^T = \psi_2$. This is not a serious problem although it does contribute to the multimodality of the likelihood.

A more serious problem is local nonidentifiability. Suppose that

$$p(x; \pi, \mu_1, \mu_2) = (1 - \pi)\phi(x; \mu_1, 1) + \pi\phi(x; \mu_2, 1). \quad (13)$$

When $\mu_1 = \mu_2 = \mu$, we see that $p(x; \pi, \mu_1, \mu_2) = \phi(x; \mu)$. The parameter π has disappeared. Similarly, when $\pi = 1$, the parameter μ_2 disappears. This means that there are subspaces of

the parameter space where the family is not identifiable. This local nonidentifiability causes many of the usual theoretical properties—such as asymptotic Normality of the maximum likelihood estimator and the limiting χ^2 behavior of the likelihood ratio test—to break down. For the model (13), there is no simple theory to describe the distribution of the likelihood ratio test for $H_0 : \mu_1 = \mu_2$ versus $H_1 : \mu_1 \neq \mu_2$. The best available theory is very complicated. However, some progress has been made lately using ideas from algebraic geometry (Yamazaki and Watanabe 2003, Watanabe 2010).

The lack of local identifiability causes other problems too. For example, we usually have that the Fisher information is non-zero and that $\hat{\theta} - \theta = O_P(n^{-1/2})$ where $\hat{\theta}$ is the maximum likelihood estimator. Mixture models are, in general, irregular: they do not satisfy the usual regularity conditions that make parametric models so easy to deal with. Here is an example from Chen (1995).

Consider a univariate mixture of two Gaussians distribution:

$$p_\theta(x) = \frac{2}{3}\phi(x; -\theta, 1) + \frac{1}{3}\phi(x; 2\theta, 1).$$

Then it is easy to check that $I(0) = 0$ where $I(\theta)$ is the Fisher information. Moreover, no estimator of θ can converge faster than $n^{-1/4}$ if the number of components is not known in advance. Compare this to a Normal family $\phi(x; \theta, 1)$ where the Fisher information is $I(\theta) = n$ and the maximum likelihood estimator converges at rate $n^{-1/2}$. Moreover, the distribution of the mle is not even well understood for mixture models. The same applies to the likelihood ratio test.

Nonintuitive Group Membership. Our motivation for studying mixture modes in this chapter was clustering. But one should be aware that mixtures can exhibit unexpected behavior with respect to clustering. Let

$$p(x) = (1 - \pi)\phi(x; \mu_1, \sigma_1^2) + \pi\phi(x; \mu_2, \sigma_2^2).$$

Suppose that $\mu_1 < \mu_2$. We can classify an observation as being from cluster 1 or cluster 2 by computing the probability of being from the first or second component, denoted $Z = 0$ and $Z = 1$. We get

$$\mathbb{P}(Z = 0|X = x) = \frac{(1 - \pi)\phi(x; \mu_1, \sigma_1^2)}{(1 - \pi)\phi(x; \mu_1, \sigma_1^2) + \pi\phi(x; \mu_2, \sigma_2^2)}.$$

Define $Z(x) = 0$ if $\mathbb{P}(Z = 0|X = x) > 1/2$ and $Z(x) = 1$ otherwise. When σ_1 is much larger than σ_2 , Figure 13 shows $Z(x)$. We end up classifying all the observations with large X_i to the leftmost component. Technically this is correct, yet it seems to be an unintended consequence of the model and does not capture what we mean by a cluster.

Improper Posteriors. Bayesian inference is based on the posterior distribution $p(\psi|X_1, \dots, X_n) \propto \mathcal{L}(\psi)\pi(\psi)$. Here, $\pi(\psi)$ is the prior distribution that represents our knowledge of ψ before

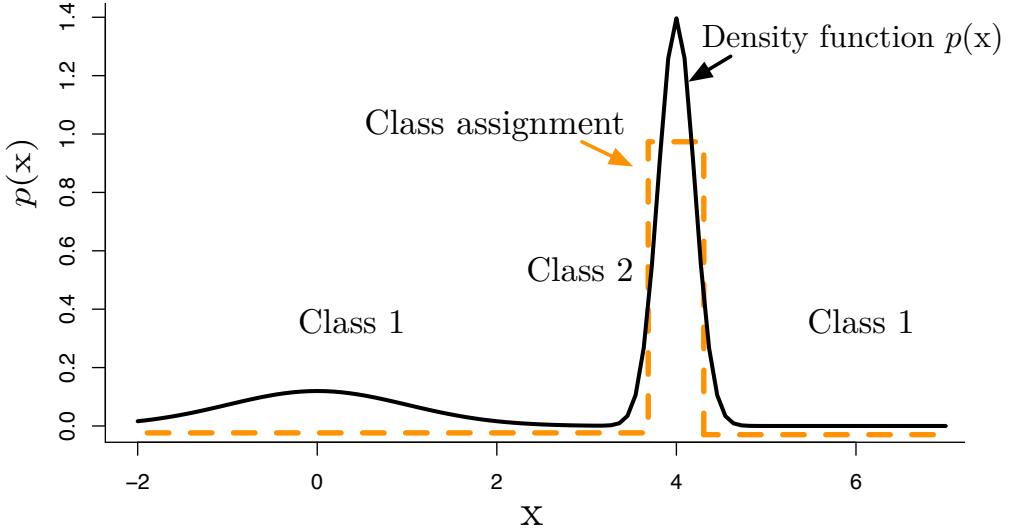


Figure 13: Mixtures are used as a parametric method for finding clusters. Observations with $x = 0$ and $x = 6$ are both classified into the first component.

seeing the data. Often, the prior is improper, meaning that it does not have a finite integral. For example, suppose that $X_1, \dots, X_n \sim N(\mu, 1)$. It is common to use an improper prior $\pi(\mu) = 1$. This is improper because

$$\int \pi(\mu) d\mu = \infty.$$

Nevertheless, the posterior $p(\mu|\mathcal{D}_n) \propto \mathcal{L}(\mu)\pi(\mu)$ is a proper distribution, where $\mathcal{L}(\mu)$ is the data likelihood of μ . In fact, the posterior for μ is $N(\bar{x}, 1/\sqrt{n})$ where \bar{x} is the sample mean. The posterior inferences in this case coincide exactly with the frequentist inferences. In many parametric models, the posterior inferences are well defined even if the prior is improper and usually they approximate the frequentist inferences. Not so with mixtures. Let

$$p(x; \mu) = \frac{1}{2}\phi(x; 0, 1) + \frac{1}{2}\phi(x; \mu, 1). \quad (14)$$

If $\pi(\mu)$ is improper then so is the posterior. Moreover, Wasserman (2000) shows that the only priors that yield posteriors in close agreement to frequentist methods are data-dependent priors.

Use With Caution. Mixture models can have very unusual and unexpected behavior. This does not mean that we should not use mixture models. Indeed, mixture models are extremely useful. However, when you use mixture models, it is important to keep in mind that many of the properties of models that we often take for granted, may not hold.

What Does All This Mean? Mixture models can have very unusual and unexpected behavior. This does not mean that we should not use mixture models. Compare this to

kernel density estimators which are simple and very well understood. If you are going to use mixture models, I advise you to remember the words of Rod Serling:

There is a fifth dimension beyond that which is known to man. It is a dimension as vast as space and as timeless as infinity. It is the middle ground between light and shadow, between science and superstition, and it lies between the pit of man's fears and the summit of his knowledge. This is the dimension of imagination. It is an area which we call the Twilight Zone.

4 Density-Based Clustering I: Level Set Clustering

Let p be the density if the data. Let $L_t = \{x : p_h(x) > t\}$ denote an upper level set of p . Suppose that L_t can be decomposed into finitely many disjoint sets: $L_t = C_1 \cup \dots \cup C_{k_t}$. We call $\mathcal{C}_t = \{C_1, \dots, C_{k_t}\}$ the level set clusters at level t .

Let $\mathcal{C} = \bigcup_{t \geq 0} \mathcal{C}_t$. The clusters in \mathcal{C} form a tree: if $A, B \in \mathcal{C}$, the either (i) $A \subset B$ or (ii) $B \subset A$ or (iii) $A \cap B = \emptyset$. We call \mathcal{C} the *level set cluster tree*.

The level sets can be estimated in the obvious way: $\widehat{L}_t = \{x : \widehat{p}_h(x) > t\}$. How do we decompose \widehat{L}_t into its connected components? This can be done as follows. For each t let

$$\mathcal{X}_t = \{X_i : \widehat{p}_h(X_i) > t\}.$$

Now construct a graph G_t where each $X_i \in \mathcal{X}_t$ is a vertex and there is an edge between X_i and X_j if and only if $\|X_i - X_j\| \leq \epsilon$ where $\epsilon > 0$ is a tuning parameter. Bobrowski et al (2014) show that we can take $\epsilon = h$. G_t is called a Rips graphs. The clusters at level t are estimated by taking the connected components of the graph G_t . In summary:

1. Compute \widehat{p}_h .
2. For each t , let $\mathcal{X}_t = \{X_i : \widehat{p}_h(X_i) > t\}$.
3. Form a graph G_t for the points in \mathcal{X}_t by connecting X_i and X_j if $\|X_i - X_j\| \leq h$.
4. The clusters at level t are the connected components of G_t .

A Python package, called DeBaCl, written by Brian Kent, can be found at

<http://www.briankent.com/projects.html>.

Fabrizio Lecci has written an R implementation, included in his R package: TDA (topological data analysis). You can get it at:

<http://cran.r-project.org/web/packages/TDA/index.html>

Two examples are shown in Figures 14 and 15.

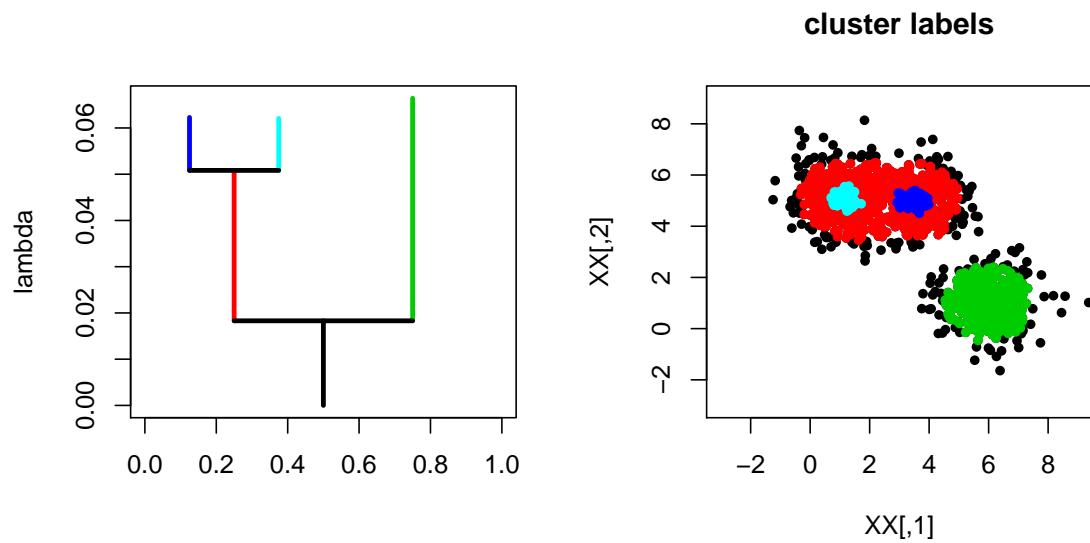


Figure 14: DeBaCLR in two dimensions.

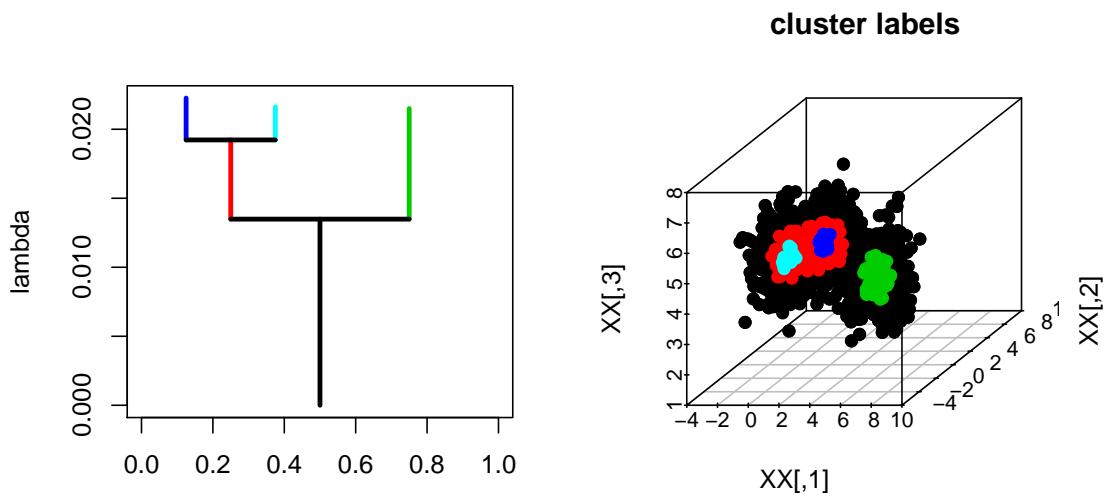


Figure 15: DeBaCLR in three dimensions.

4.1 Theory

How well does this work? Define the Hausdorff distance between two sets by

$$H(U, V) = \inf \left\{ \epsilon : U \subset V \oplus \epsilon \text{ and } V \subset U \oplus \epsilon \right\}$$

where

$$V \oplus \epsilon = \bigcup_{x \in V} B(x, \epsilon)$$

and $B(x, \epsilon)$ denotes a ball of radius ϵ centered at x . We would like to say that L_t and \widehat{L}_t are close. In general this is not true. Sometimes L_t and $L_{t+\delta}$ are drastically different even for small δ . (Think of the case where a mode has height t .) But we can estimate stable level sets. Let us say that L_t is stable if there exists $a > 0$ and $C > 0$ such that, for all $\delta < a$,

$$H(L_{t-\delta}, L_{t+\delta}) \leq C\delta.$$

Theorem 11 Suppose that L_t is stable. Then $H(\widehat{L}_t, L_t) = O_P(\sqrt{\log n/(nh^d)})$.

Proof. Let $r_n = \sqrt{\log n/(nh^d)}$. We need to show two things: (i) for every $x \in L_t$ there exists $y \in \widehat{L}_t$ such that $\|x - y\| = O_P(r_n)$ and (ii) for every $x \in \widehat{L}_t$ there exists $y \in L_t$ such that $\|x - y\| = O_P(r_n)$. First, we note that, by earlier results, $\|\widehat{p}_h - p_h\|_\infty = O_P(r_n)$. To show (i), suppose that $x \in L_t$. By the stability assumption, there exists $y \in L_{t+r_n}$ such that $\|x - y\| \leq Cr_n$. Then $p_h(y) > t + r_n$ which implies that $\widehat{p}_h(y) > t$ and so $y \in \widehat{L}_t$. To show (ii), let $x \in \widehat{L}_t$ so that $\widehat{p}_h(x) > t$. Thus $p_h(x) > t - r_n$. By stability, there is a $y \in L_t$ such that $\|x - y\| \leq Cr_n$. \square

4.2 Persistence

Consider a smooth density p with $M = \sup_x p(x) < \infty$. The t -level set clusters are the connected components of the set $L_t = \{x : p(x) \geq t\}$. Suppose we find the upper level sets $L_t = \{x : p(x) \geq t\}$ as we vary t from M to 0. *Persistent homology* measures how the topology of L_t varies as we decrease t . In our case, we are only interested in the modes, which correspond to the zeroth order homology. (Higher order homology refers to holes, tunnels etc.) The idea of using persistence to study clustering was introduced by Chazal, Guibas, Oudot and Skraba (2013).

Imagine setting $t = M$ and then gradually decreasing t . Whenever we hit a mode, a new level set cluster is born. As we decrease t further, some clusters may merge and we say that one of the clusters (the one born most recently) has died. See Figure 16.

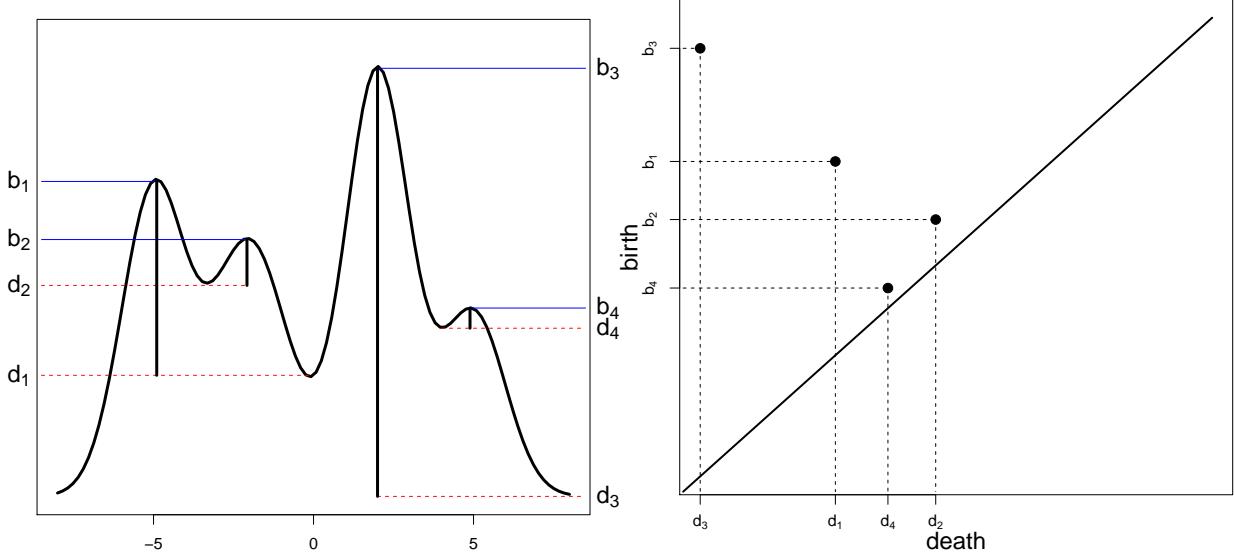


Figure 16: Starting at the top of the density and moving down, each mode has a birth time b and a death time d . The persistence diagram (right) plots the points $(d_1, b_1), \dots, (d_4, b_4)$. Modes with a long lifetime are far from the diagonal.

In summary, each mode m_j has a death time and a birth time denoted by (d_j, b_j) . (Note that the birth time is larger than the death time because we start at high density and move to lower density.) The modes can be summarized with a persistence diagram where we plot the points $(d_1, b_1), \dots, (d_k, b_k)$ in the plane. See Figure 16. Points near the diagonal correspond to modes with short lifetimes. We might kill modes with lifetimes smaller than the bootstrap quantile ϵ_α defined by

$$\epsilon_\alpha = \inf \left\{ z : \frac{1}{B} \sum_{b=1}^B I \left(\|\hat{p}_h^{*b} - \hat{p}_h\|_\infty > z \right) \leq \alpha \right\}. \quad (15)$$

Here, \hat{p}_h^{*b} is the density estimator based on the b^{th} bootstrap sample. This corresponds to killing a mode if it is in a $2\epsilon_\alpha$ band around the diagonal. See Fasy, Lecci, Rinaldo, Wasserman, Balakrishnan and Singh (2014). Note that the starting and ending points of the vertical bars on the level set tree are precisely the coordinates of the persistence diagram. (A more precise bootstrap approach was introduced in Chazal, Fasy, Lecci, Michel, Rinaldo and Wasserman (2104).)

5 Density-Based Clustering II: Modes

Let p be the density of $X \in \mathbb{R}^d$. Assume that p has modes m_1, \dots, m_{k_0} and that p is a *Morse function*, which means that the Hessian of p at each stationary point is non-degenerate. We

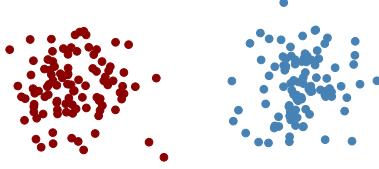


Figure 17: A synthetic example with two “blob-like” clusters.

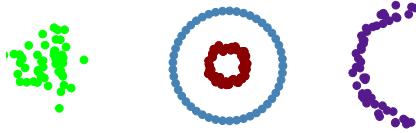


Figure 18: A synthetic example with four clusters with a variety of different shapes.

can use the modes to define clusters as follows.

5.1 Mode Clustering

Given any point $x \in \mathbb{R}^d$, there is a unique gradient ascent path, or integral curve, passing through x that eventually leads to one of the modes. We define the clusters to be the “basins of attraction” of the modes, the equivalence classes of points whose ascent paths lead to the same mode. Formally, an *integral curve* through x is a path $\pi_x : \mathbb{R} \rightarrow \mathbb{R}^d$ such that $\pi_x(0) = x$ and

$$\pi'_x(t) = \nabla p(\pi_x(t)). \quad (16)$$

Integral curves never intersect (except at stationary points) and they partition the space.

Equation (16) means that the path π follows the direction of steepest ascent of p through x . The destination of the integral curve π through a (non-mode) point x is defined by

$$\text{dest}(x) = \lim_{t \rightarrow \infty} \pi_x(t). \quad (17)$$

It can then be shown that for all x , $\text{dest}(x) = m_j$ for some mode m_j . That is: all integral curves lead to modes. For each mode m_j , define the sets

$$\mathcal{A}_j = \left\{ x : \text{dest}(x) = m_j \right\}. \quad (18)$$

These sets are known as the *ascending manifolds*, and also known as the cluster associated with m_j , or the basin of attraction of m_j . The \mathcal{A}_j ’s partition the space. See Figure 19. The collection of ascending manifolds is called the *Morse complex*.

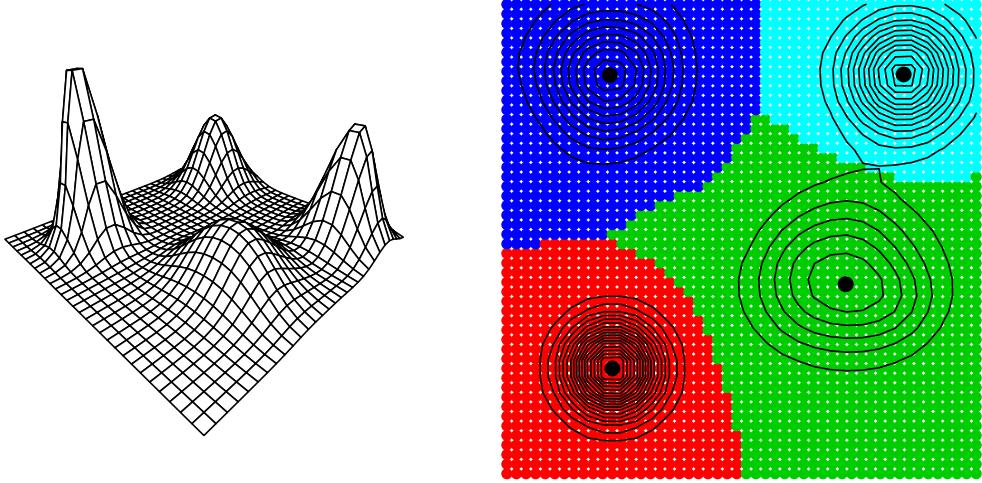


Figure 19: The left plot shows a function with four modes. The right plot shows the ascending manifolds (basins of attraction) corresponding to the four modes.

Given data X_1, \dots, X_n we construct an estimate \hat{p} of the density. Let $\hat{m}_1, \dots, \hat{m}_k$ be the estimated modes and let $\hat{\mathcal{A}}_1, \dots, \hat{\mathcal{A}}_k$ be the corresponding ascending manifolds derived from \hat{p} . The sample clusters C_1, \dots, C_k are defined to be $C_j = \{X_i : X_i \in \hat{\mathcal{A}}_j\}$.

Recall that the kernel density estimator is

$$\hat{p}(x) \equiv \hat{p}_h(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^d} K\left(\frac{\|x - X_i\|}{h}\right) \quad (19)$$

where K is a smooth, symmetric kernel and $h > 0$ is the bandwidth.¹ The mean of the estimator is

$$p_h(x) = \mathbb{E}[\hat{p}_h(x)] = \int K(t)p(x + th)dt. \quad (20)$$

To locate the modes of \hat{p}_h we use the *mean shift algorithm* which finds modes by approximating the steepest ascent paths. The algorithm is given in Figure 20. The result of this process is the set of estimated modes $\widehat{\mathcal{M}} = \{\hat{m}_1, \dots, \hat{m}_k\}$. We also get the clustering for free: the mean shift algorithm shows us what mode each point is attracted to. See Figure 21.

A modified version of the algorithm is the blurred mean-shift algorithm (Carreira-Perpinan, 2006). Here, we use the data as the mesh and we replace the data with the mean-shifted data at each step. This converges very quickly but must be stopped before everything converges to a single point; see Figures 22 and 23.

¹In general, we can use a bandwidth matrix H in the estimator, with $\hat{p}(x) \equiv \hat{p}_H(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - X_i)$ where $K_H(x) = |H|^{-\frac{1}{2}} K(H^{-\frac{1}{2}}x)$.

Mean Shift Algorithm

1. Input: $\hat{p}(x)$ and a mesh of points $A = \{a_1, \dots, a_N\}$ (often taken to be the data points).
2. For each mesh point a_j , set $a_j^{(0)} = a_j$ and iterate the following equation until convergence:
$$a_j^{(s+1)} \leftarrow \frac{\sum_{i=1}^n X_i K \left(\frac{\|a_j^{(s)} - X_i\|}{h} \right)}{\sum_{i=1}^n K \left(\frac{\|a_j^{(s)} - X_i\|}{h} \right)}.$$
3. Let $\widehat{\mathcal{M}}$ be the unique values of the set $\{a_1^{(\infty)}, \dots, a_N^{(\infty)}\}$.
4. Output: $\widehat{\mathcal{M}}$.

Figure 20: *The Mean Shift Algorithm.*

What we are doing is tracing out the *gradient flow*. The flow lines lead to the modes and they define the clusters. In general, a flow is a map $\phi : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ such that $\phi(x, 0) = x$ and $\phi(\phi(x, t), s) = \phi(x, s + t)$. The latter is called the semi-group property.

5.2 Choosing the Bandwidth

As usual, choosing a good bandwidth is crucial. You might wonder if increasing the bandwidth, decreases the number of modes. Silverman (1981) showed that the answer is yes if you use a Normal kernel.

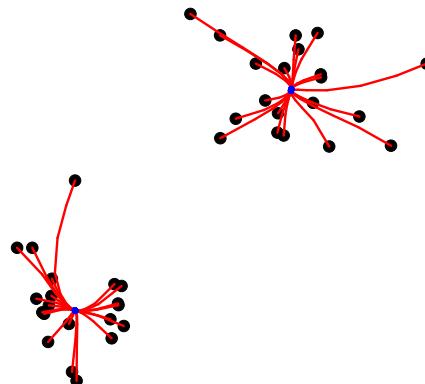


Figure 21: A simple example of the mean shift algorithm.

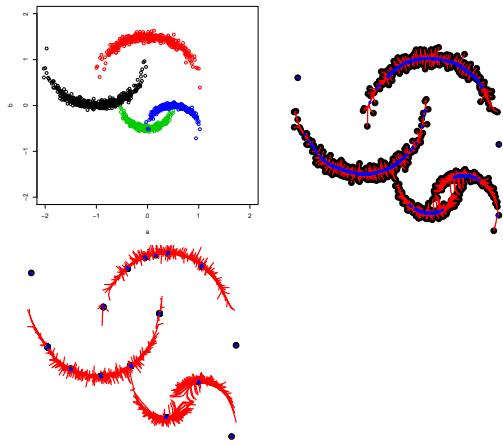


Figure 22: The crescent data example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

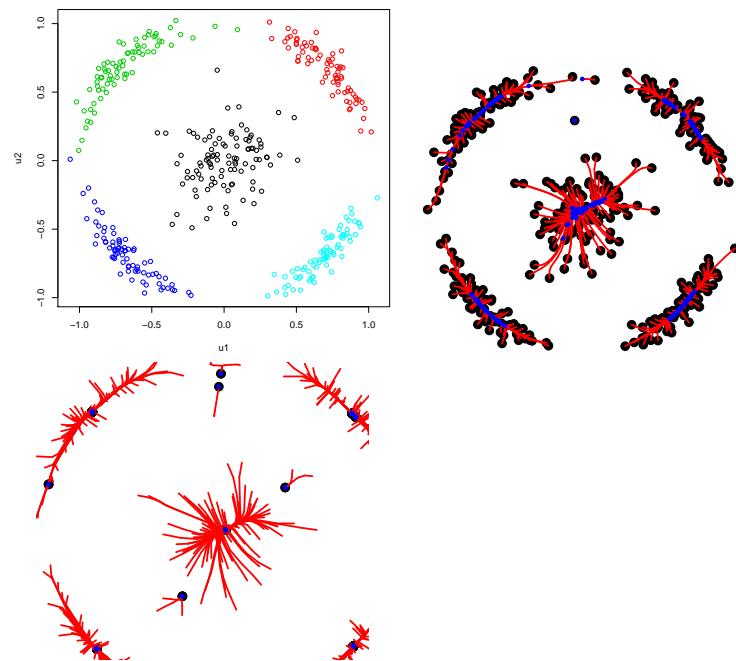


Figure 23: The Broken Ring example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

Theorem 12 (Silverman 1981) Let \hat{p}_h be a kernel density estimator using a Gaussian kernel in one dimension. Then the number of modes of \hat{p}_h is a non-increasing function of h . The Gaussian kernel is the unique kernel with this property.

We still need a way to pick h . We can use cross-validation as before. One could argue that we should choose h so that we estimate the gradient $g(x) = \nabla p(x)$ well since the clustering is based on the gradient flow.

How can we estimate the loss of the gradient? Consider, first the scalar case. Note that

$$\int (\hat{p}' - p')^2 = \int (\hat{p}')^2 - 2 \int \hat{p} p' + \int (p')^2.$$

We can ignore the last term. The first term is known. To estimate the middle term, we use integration by parts to get

$$\int \hat{p} p' = - \int p'' p$$

suggesting the cross-validation estimator

$$\int (\hat{p}'(x))^2 dx + \frac{2}{n} \sum_i \hat{p}_i''(X_i)$$

where \hat{p}_i'' is the leave-one-out second derivative. More generally, by repeated integration by parts, we can estimate the loss for the r^{th} derivative by

$$\text{CV}_r(h) = \int (\hat{p}^{(r)}(x))^2 dx - \frac{2}{n} (-1)^r \sum_i \hat{p}_i^{(2r)}(X_i).$$

Let's now discuss estimating derivatives more generally following Chacon and Duong (2013). Let

$$\hat{p}_H(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - X_i)$$

where $K_H(x) = |H|^{-1/2} K(H^{-1/2}x)$. Let $D = \partial/\partial x = (\partial/\partial x_1, \dots, \partial/\partial x_d)$ be the gradient operator. Let $H(x)$ be the Hessian of $p(x)$ whose entries are $\partial^2 p / (\partial x_j \partial x_k)$. Let

$$D^{\otimes r} p = (Dp)^{\otimes r} = \partial^r p / \partial x^{\otimes r} \in \mathbb{R}^{d^r}$$

denote the r^{th} derivatives, organized into a vector. Thus

$$D^{\otimes 0} p = p, \quad D^{\otimes 1} p = Dp, \quad D^{\otimes 2} p = \text{vec}(H)$$

where vec takes a matrix and stacks the columns into a vector.

The estimate of $D^{\otimes r}p$ is

$$\widehat{p}^{(r)}(x) = D^{\otimes r}\widehat{p}_H(x) = \frac{1}{n} \sum_{i=1}^n D^{\otimes r}K_H(x-X_i) = \frac{1}{n} \sum_{i=1}^n |H|^{-1/2}(H^{-1/2})^{\otimes r}D^{\otimes r}K(H^{-1/2}(x-X_i)).$$

The integrated squared error is

$$L = \int ||D^{\otimes r}\widehat{p}_H(x) - D^{\otimes r}p(x)||^2 dx.$$

Chacon, Duong and Wand shows that $\mathbb{E}[L]$ is minimized by choosing H so that each entry has order $n^{-2/(d+2r+4)}$ leading to a risk of order $O(n^{-4/(d+2r+4)})$. In fact, it may be shown that

$$\begin{aligned} \mathbb{E}[L] &= \frac{1}{n}|H|^{-1/2}\text{tr}((H^{-1})^{\otimes r}R(D^{\otimes r}K)) - \frac{1}{n}\text{tr}R^*(K_H \star K_H, D^{\otimes r}p) \\ &\quad + \text{tr}R^*(K_H \star K_H, D^{\otimes r}p) - 2\text{tr}R^*(K_H, D^{\otimes r}p) + \text{tr}R(D^{\otimes r}p) \end{aligned}$$

where

$$\begin{aligned} R(g) &= \int g(x)g^T(x)dx \\ R^*(a, g) &= \int (a \star g)(x)g^T(x)dx \end{aligned}$$

and $(a \star g)$ is componentwise convolution.

To estimate the loss, we expand L as

$$L = \int ||D^{\otimes r}\widehat{p}_H(x)||^2 dx - 2 \int \langle D^{\otimes r}\widehat{p}_H(x), D^{\otimes r}p(x) \rangle dx + \text{constant}.$$

Using some high-voltage calculations, Chacon and Duong (2013) derived the following leave-one-out approximation to the first two terms:

$$\text{CV}_r(H) = (-1)^r |H|^{-1/2} (\text{vec}(H^{-1})^{\otimes r})^T B(H)$$

where

$$B(H) = \frac{1}{n^2} \sum_{i,j} D^{\otimes 2r} \overline{K}(H^{-1/2}(X_i - X_j)) - \frac{2}{n(n-1)} \sum_{i \neq j} D^{\otimes 2r} K(H^{-1/2}(X_i - X_j))$$

and $\overline{K} = K \star K$. In practice, the minimization is easy if we restrict to matrices of the form $H = h^2 I$.

A better idea is to used fixed (non-decreasing h). We don't need h to go to 0 to find the clusters. More on this when we discuss persistence.

5.3 Theoretical Analysis

How well can we estimate the modes?

Theorem 13 Assume that p is Morse with finitely many modes m_1, \dots, m_k . Then for $h > 0$ and not too large, p_h is Morse with modes m_{h1}, \dots, m_{hk} and (possibly after relabelling),

$$\max_j \|m_j - m_{jh}\| = O(h^2).$$

With probability tending to 1, \hat{p}_h has the same number of modes which we denote by $\hat{m}_{h1}, \dots, \hat{m}_{hk}$. Furthermore,

$$\max_j \|\hat{m}_{jh} - m_{jh}\| = O_P \left(\sqrt{\frac{1}{nh^{d+2}}} \right)$$

and

$$\max_j \|\hat{m}_{jh} - m_j\| = O(h^2) + O_P \left(\sqrt{\frac{1}{nh^{d+2}}} \right).$$

Remark: Setting $h \asymp n^{-1/(d+6)}$ gives the rate $n^{-2/(d+6)}$ which is minimax (Tsyabkov 1990) under smoothness assumptions. See also Romano (1988). However, if we take the fixed h point of view, then we have a $n^{-1/2}$ rate.

Proof Outline. But a small ball B_j around each m_{jh} . We will skip the first step, which is to show that there is one (and only one) local mode in B_j . Let's focus on showing

$$\max_j \|\hat{m}_{jh} - m_{jh}\| = O_P \left(\sqrt{\frac{1}{nh^{d+2}}} \right).$$

For simplicity, write $m = m_{jh}$ and $x = \hat{m}_{jh}$. Let $g(x)$ and $H(x)$ be the gradient and Hessian of $p_h(x)$ and let $\hat{g}(x)$ and $\hat{H}(x)$ be the gradient Hessian of $\hat{p}_h(x)$. Then

$$(0, \dots, 0)^T = \hat{g}(x) = \hat{g}(m) + (x - m)^T \int_0^1 \hat{H}(m + u(x - m)) du$$

and so

$$(x - m)^T \int_0^1 \hat{H}(m + u(x - m)) du = (g(m) - \hat{g}(m))$$

where we used the fact that $\mathbf{0} = g(m)$. Multiplying on the right by $x - m$ we have

$$(x - m)^T \int_0^1 \hat{H}(m + u(x - m))(x - m) du = (\hat{g}(m) - \hat{g}(m))^T (x - m).$$

Let $\lambda = \inf_{0 \leq u \leq 1} \lambda_{\min}(H(m + u(x - m)))$. Then $\lambda = \lambda_{\min}(H(m)) + o_P(1)$ and

$$(x - m)^T \int_0^1 \widehat{H}(x + u(m - x))(x - m) du \geq \lambda \|x - m\|^2.$$

Hence, using Cauchy-Schwartz,

$$\lambda \|x - m\|^2 \leq \|\widehat{g}(m) - g(m)\| \|x - m\| \leq \|x - m\| \sup_y \|\widehat{g}(y) - \widehat{g}(y)\| \leq \|x - m\| O_P \left(\sqrt{\frac{1}{nh^{d+2}}} \right)$$

and so $\|x - m\| = O_P \left(\sqrt{\frac{1}{nh^{d+2}}} \right)$. \square

Remark: If we treat h as fixed (not decreasing) then the rate is $O_P(\sqrt{1/n})$ independent of dimension.

6 Hierarchical Clustering

Hierarchical clustering methods build a set of nested clusters at different resolutions. There are two types of hierarchical clustering: agglomerative (bottom-up) and divisive (top-down). With agglomerative clustering we start with some distance or dissimilarity $d(x, y)$ between points. We then extend this distance so that we can compute the distance $d(A, B)$ between two sets of points A and B .

The three most common ways of extending the distance are:

Single Linkage	$d(A, B) = \min_{x \in A, y \in B} d(x, y)$
Average Linkage	$d(A, B) = \frac{1}{N_A N_B} \sum_{x \in A, y \in B} d(x, y)$
Complete Linkage	$d(A, B) = \max_{x \in A, y \in B} d(x, y)$

The algorithm is:

1. Input: data $X = \{X_1, \dots, X_n\}$ and metric d giving distance between clusters.
2. Let $T_n = \{C_1, C_2, \dots, C_n\}$ where $C_i = \{X_i\}$.
3. For $j = n - 1$ to 1:

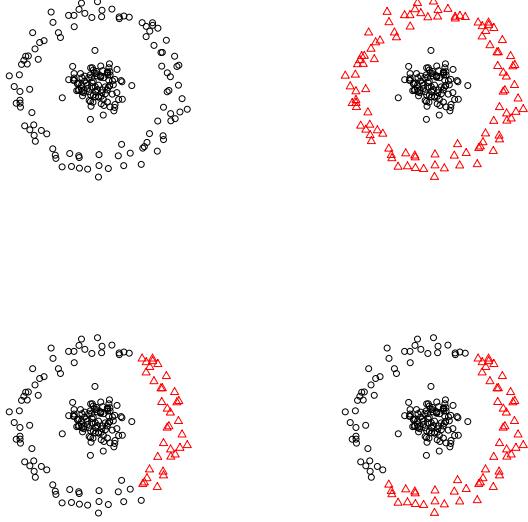


Figure 24: Hierarchical clustering applied to two noisy rings. Top left: the data. Top right: two clusters from hierarchical clustering using single linkage. Bottom left: average linkage. Bottom right: complete linkage.

- (a) Find j, k to minimize $d(C_j, C_k)$ over all $C_j, C_k \in T_{j+1}$.
- (b) Let T_j be the same as T_{j+1} except that C_j and C_k are replaced with $C_j \cup C_k$.
- 4. Return the sets of clusters T_1, \dots, T_n .

The result can be represented as a tree, called a dendrogram. We can then cut the tree at different places to yield any number of clusters ranging from 1 to n . Single linkage often produces thin clusters while complete linkage is better at rounder clusters. Average linkage is in between.

Example 14 Figure 24 shows agglomerative clustering applied to data generated from two rings plus noise. The noise is large enough so that the smaller ring looks like a blob. The data are shown in the top left plot. The top right plot shows hierarchical clustering using single linkage. (The tree is cut to obtain two clusters.) The bottom left plot shows average linkage and the bottom right plot shows complete linkage. Single linkage works well while average and complete linkage do poorly.

Let us now mention some theoretical properties of hierarchical clustering. Suppose that X_1, \dots, X_n is a sample from a distribution P on \mathbb{R}^d with density p . A high density cluster is a maximal connected component of a set of the form $\{x : p(x) \geq \lambda\}$. One might expect that single linkage clusters would correspond to high density clusters. This turns out not quite to be the case. See Hartigan (1981) for details. DasGupta (2010) has a modified version

of hierarchical clustering that attempts to fix this problem. His method is very similar to density clustering.

Single linkage hierarchical clustering is the same as *geometric graph clustering*. Let $G = (V, E)$ be a graph where $V = \{X_1, \dots, X_n\}$ and $E_{ij} = 1$ if $\|X_i - X_j\| \leq \epsilon$ and $E_{ij} = 0$ if $\|X_i - X_j\| > \epsilon$. Let C_1, \dots, C_k denote the connected components of the graph. As we vary ϵ we get exactly the hierarchical clustering tree.

Finally, we let us mention divisive clustering. This is a form of hierarchical clustering where we start with one large cluster and then break the cluster recursively into smaller and smaller pieces.

7 Spectral Clustering

Spectral clustering refers to a class of clustering methods that use ideas related to eigenvector. An excellent tutorial on spectral clustering is von Luxburg (2006) and some of this section relies heavily on that paper. More detail can be found in Chung (1997).

Let G be an undirected graph with n vertices. Typically these vertices correspond to observations X_1, \dots, X_n . Let W be an $n \times n$ symmetric weight matrix. Say that X_i and X_j are connected if $W_{ij} > 0$. The simplest type of weight matrix has entries that are either 0 or 1. For example, we could define

$$W_{ij} = I(\|X_i - X_j\| \leq \epsilon).$$

An example of a more general weight matrix is $W_{ij} = e^{-\|X_i - X_j\|^2/(2h^2)}$.

The degree matrix D is the $n \times n$ diagonal matrix with $D_{ii} = \sum_{j=1}^n W_{ij}$. The graph Laplacian is

$$L = D - W. \tag{21}$$

The graph Laplacian has many interesting properties which we list in the following result. Recall that a vector v is an eigenvector of L if there is a scalar λ such that $Lv = \lambda v$ in which case we say that λ is the eigenvalue corresponding to v . Let $\mathcal{L}(v) = \{cv : c \in \mathbb{R}, c \neq 0\}$ be the linear space generated by v . If v is an eigenvector with eigenvalue λ and c is any nonzero constant, then cv is an eigenvector with eigenvalue $c\lambda$. These eigenvectors are considered equivalent. In other words, $\mathcal{L}(v)$ is the set of vectors that are equivalent to v .

Theorem 15 *The graph Laplacian L has the following properties:*

1. For any vector $f = (f_1, \dots, f_n)^T$,

$$f^T L f = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} (f_i - f_j)^2.$$

2. L is symmetric and positive semi-definite.

3. The smallest eigenvalue of L is 0. The corresponding eigenvector is $(1, 1, \dots, 1)^T$.

4. L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_k$.

5. The number of eigenvalues that are equal to 0 is equal to the number of connected components of G . That is, $0 = \lambda_1 = \dots = \lambda_k$ where k is the number of connected components of G . The corresponding eigenvectors v_1, \dots, v_k are orthogonal and each is constant over one of the connected components of the graph.

Part 1 of the theorem says that L is like a derivative operator. The last part shows that we can use the graph Laplacian to find the connected components of the graph.

Proof.

(1) This follows from direct algebra.

(2) Since W and D are symmetric, it follows that L is symmetric. The fact that L is positive semi-definite follows from part (1).

(3) Let $v = (1, \dots, 1)^T$. Then

$$Lv = Dv - Wv = \begin{pmatrix} D_{11} \\ \vdots \\ D_{nn} \end{pmatrix} - \begin{pmatrix} D_{11} \\ \vdots \\ D_{nn} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

which equals $0 \times v$.

(4) This follows from parts (1)-(3).

(5) First suppose that $k = 1$ and thus that the graph is fully connected. We already know that $\lambda_1 = 0$ and $v_1 = (1, \dots, 1)^T$. Suppose there were another eigenvector v with eigenvalue 0. Then

$$0 = v^T Lv = \sum_{i=1}^n \sum_{j=1}^n W_{ij} (v(i) - v(j))^2.$$

It follows that $W_{ij}(v(i) - v(j))^2 = 0$ for all i and j . Since G is fully connected, all $W_{ij} > 0$. Hence, $v(i) = v(j)$ for all i, j and so v is constant and thus $v \in \mathcal{L}(v_1)$.

Now suppose that K has k components. Let n_j be the number of nodes in component j . We can re-label the vertices so that the first n_1 nodes correspond to the first connected component, the second n_2 nodes correspond to the second connected component and so on. Let $v_1 = (1, \dots, 1, 0, \dots, 0)$ where the 1's correspond to the first component. Let $v_2 = (0, \dots, 0, 1, \dots, 1, 0, \dots, 0)$ where the 1's correspond to the second component. Define v_3, \dots, v_k similarly. Due to the re-ordering of the vertices, L has block diagonal form:

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}.$$

Here, each L_i corresponds to one of the connected components of the graph. It is easy to see that $LV_j = j$ for $j = 1, \dots, k$. Thus, each v_j , for $j = 1, \dots, k$ is an eigenvector with zero eigenvalue. Suppose that v is any eigenvector with 0 eigenvalue. Arguing as before, v must be constant over some component and 0 elsewhere. Hence, $v \in \mathcal{L}(v_j)$ for some $1 \leq j \leq k$. \square

Example 16 Consider the graph

$$X_1 \text{ ————— } X_2 \quad X_3 \text{ ————— } X_4 \text{ ————— } X_5$$

and suppose that $W_{ij} = 1$ if and only if there is an edge between X_i and X_j . Then

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and the Laplacian is

$$L = D - W = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}.$$

The eigenvalues of W , from smallest to largest are $0, 0, 1, 2, 3$. The eigenvectors are

$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ -.71 \\ 0 \\ .71 \end{pmatrix} \quad v_4 = \begin{pmatrix} -.71 \\ .71 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad v_5 = \begin{pmatrix} 0 \\ 0 \\ -.41 \\ .82 \\ -.41 \end{pmatrix}$$

Note that the first two eigenvectors correspond to the connected components of the graph.

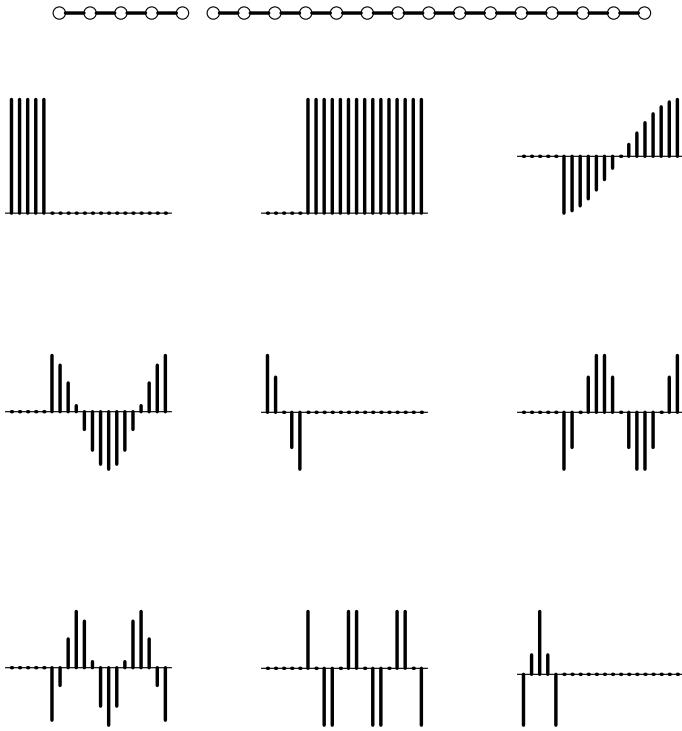


Figure 25: The top shows a simple graph. The remaining plots are the eigenvectors of the graph Laplacian. Note that the first two eigenvectors correspond to the two connected components of the graph.

Note $f^T L f$ measures the smoothness of f relative to the graph. This means that the higher order eigenvectors generate a basis where the first few basis elements are smooth (with respect to the graph) and the later basis elements become more wiggly.

Example 17 *Figure 25 shows a graph and the corresponding eigenvectors. The two eigenvectors correspond to the two connected components of the graph. The other eigenvectors can be thought of as forming basis vectors within the connected components.*

One approach to spectral clustering is to set

$$W_{ij} = I(||X_i - X_j|| \leq \epsilon)$$

for some $\epsilon > 0$ and then take the clusters to be the connected components of the graph which can be found by getting the eigenvectors of the Laplacian L . This is exactly equivalent to geometric graph clustering from Section ???. In this case we have gained nothing except that we have a new algorithm to find the connected components of the graph. However, there are other ways to use spectral methods for clustering as we now explain.

The idea underlying the other spectral methods is to use the Laplacian to transform the data into a new coordinate system in which clusters are easier to find. For this purpose, one

typically uses a modified form of the graph Laplacian. The most commonly used weights for this purpose are

$$W_{ij} = e^{-\|X_i - X_j\|^2/(2h^2)}.$$

Other kernels $K_h(X_i, X_j)$ can be used as well. We define the symmetrized Laplacian $\mathcal{L} = D^{-1/2}WD^{-1/2}$ and the random walk Laplacian $\mathcal{L} = D^{-1}W$. (We will explain the name shortly.) These are very similar and we will focus on the latter. Some authors define the random walk Laplacian to be $I - D^{-1}W$. We prefer to use the definition $\mathcal{L} = D^{-1}W$ because, as we shall see, it has a nice interpretation. The eigenvectors of $I - D^{-1}W$ and $D^{-1}W$ are the same so it makes little difference which definition is used. The main difference is that the connected components have eigenvalues 1 instead of 0.

Lemma 18 *Let L be the graph Laplacian of a graph G and let \mathcal{L} be the random walk Laplacian.*

1. λ is an eigenvalue of \mathcal{L} with eigenvector v if and only if $Lv = (1 - \lambda)Dv$.
2. 1 is an eigenvalue of \mathcal{L} with eigenvector $(1, \dots, 1)^T$.
3. \mathcal{L} is positive semidefinite with n non-negative real-valued eigenvalues.
4. The number of eigenvalues of \mathcal{L} equal to 1 equals the number of connected components of G . Let v_1, \dots, v_k denote the eigenvectors with eigenvalues equal to 1. The linear space spanned by v_1, \dots, v_k is spanned by the indicator functions of the connected components.

Proof. Homework. \square

H

Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ be the eigenvalues of \mathcal{L} with eigenvectors v_1, \dots, v_n . Define

$$Z_i \equiv T(X_i) = \sum_{j=1}^r \sqrt{\lambda_j} v_j(i).$$

The mapping $T : X \rightarrow Z$ transforms the data into a new coordinate system. The numbers h and r are tuning parameters. The hope is that clusters are easier to find in the new parameterization.

To get some intuition for this, note that \mathcal{L} has a nice probabilistic interpretation (Coifman, Lafon, Lee 2006). Consider a Markov chain on X_1, \dots, X_n where we jump from X_i to X_j with probability

$$\mathbb{P}(X_i \rightarrow X_j) = \mathcal{L}(i, j) = \frac{K_h(X_i, X_j)}{\sum_s K_h(X_s, X_j)}.$$

The Laplacian $\mathcal{L}(i, j)$ captures how easy it is to move from X_i to X_j . If Z_i and Z_j are close in Euclidean distance, then they are connected by many high density paths through the

data. This Markov chain is a discrete version of a continuous Markov chain with transition probability:

$$P(x \rightarrow A) = \frac{\int_A K_h(x, y) dP(y)}{\int K_h(x, y) dP(y)}.$$

The corresponding averaging operator $\widehat{A} : f \rightarrow \tilde{f}$ is

$$(\widehat{A}f)(i) = \frac{\sum_j f(j) K_h(X_i, X_j)}{\sum_j K_h(X_i, X_j)}$$

which is an estimate of $A : f \rightarrow \tilde{f}$ where

$$Af = \frac{\int_A f(y) K_h(x, y) dP(y)}{\int K_h(x, y) dP(y)}.$$

The lower order eigenvectors of \mathcal{L} are vectors that are smooth relative to P . Thus, projecting onto the first few eigenvectors parameterizes in terms of closeness with respect to the underlying density.

The steps are:

Input: $n \times n$ similarity matrix W .

1. Let D be the $n \times n$ diagonal matrix with $D_{ii} = \sum_j W_{ij}$.
2. Compute the Laplacian $\mathcal{L} = D^{-1}W$.
3. Find first k eigenvectors v_1, \dots, v_k of \mathcal{L} .
4. Project each X_i onto the eigenvectors to get new points \widehat{X}_i .
5. Cluster the points $\widehat{X}_1, \dots, \widehat{X}_n$ using any standard clustering algorithm.

There is another way to think about spectral clustering. Spectral methods are similar to multidimensional scaling. However, multidimensional scaling attempts to reduce dimension while preserving all pairwise distances. Spectral methods attempt instead to preserve local distances.

Example 19 Figure 26 shows a simple synthetic example. The top left plot shows the data. We apply spectral clustering with Gaussian weights and bandwidth $h = 3$. The top middle plot shows the first 20 eigenvalues. The top right plot shows the the first versus the second eigenvector. The two clusters are clearly separated. (Because the clusters are so separated, the graph is essentially disconnected and the first eigenvector is not constant. For large h , the graph becomes fully connected and v_1 is then constant.) The remaining six plots show the first six eigenvectors. We see that they form a Fourier-like basis within each cluster. Of course, single linkage clustering would work just as well with the original data as in the transformed data. The real advantage would come if the original data were high dimensional.

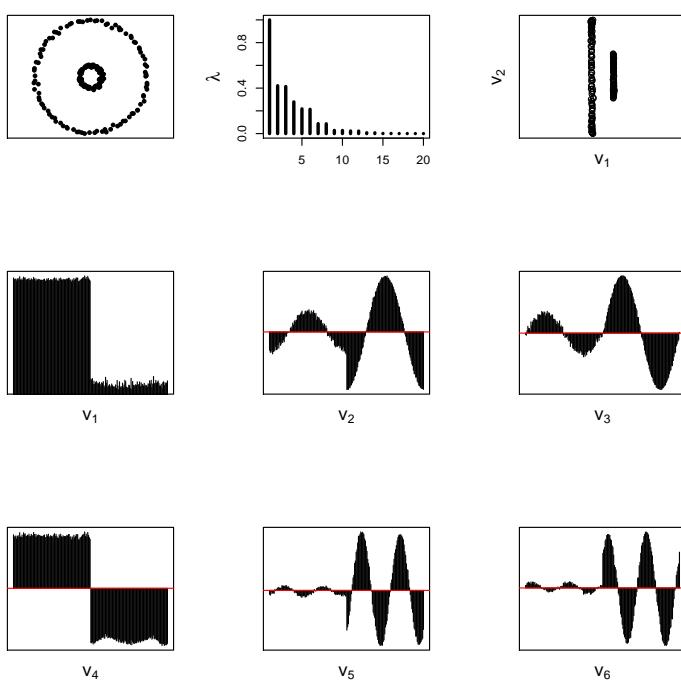


Figure 26: Top left: data. Top middle: eigenvalues. Top right: second versus third eigenvectors. Remaining plots: first six eigenvectors.

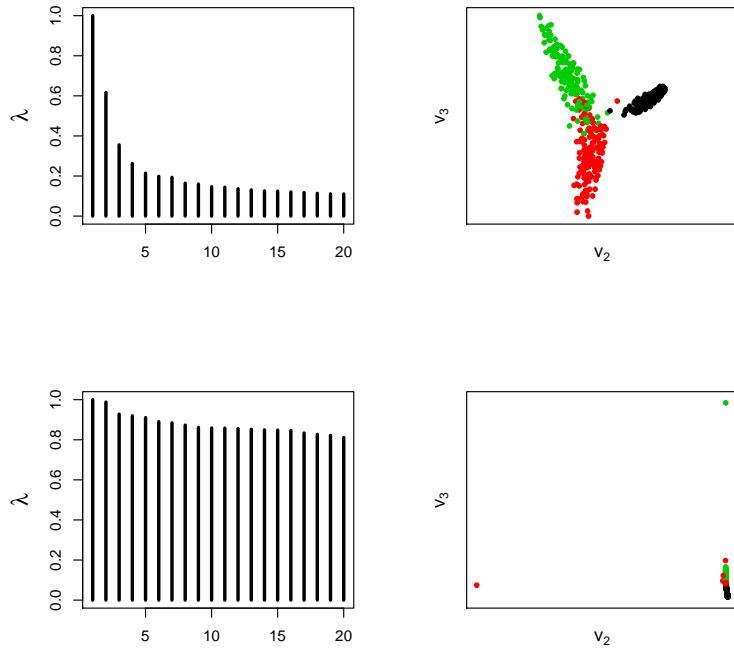


Figure 27: Spectral analysis of some zipcode data. Top: $h = 6$. Bottom: $h = 4$. The plots on the right show the second versus third eigenvector. The three colors correspond to the three digits 1, 2 and 3.

Example 20 *Figure 27 shows a spectral analysis of some zipcode data. Each datapoint is a 16×16 image of a handwritten number. We restrict ourselves to the digits 1, 2 and 3. We use Gaussian weights and the top plots correspond to $h = 6$ while the bottom plots correspond to $h = 4$. The left plots show the first 20 eigenvalues. The right plots show a scatterplot of the second versus the third eigenvector. The three colors correspond to the three digits. We see that with a good choice of h , namely $h = 6$, we can clearly see the digits in the plot. The original dimension of the problem is $16 \times 16 = 256$. That is, each image can be represented by a point in \mathbb{R}^{256} . However, the spectral method shows that most of the information is captured by two eigenvectors so the effective dimension is 2. This example also shows that the choice of h is crucial.*

Spectral methods are interesting. However, there are some open questions:

1. There are tuning parameters (such as h) and the results are sensitive to these parameters. How do we choose these tuning parameters?
2. Does spectral clustering perform better than density clustering?

8 High-Dimensional Clustering

As usual, interesting and unexpected things happen in high dimensions. The usual methods may break down and even the meaning of a cluster may not be clear.

8.1 High Dimensional Behavior

I'll begin by discussing some recent results from Sarkar and Ghosh (arXiv:1612.09121). Suppose we have data coming from k distributions P_1, \dots, P_k . Let μ_r be the mean of P_r and Σ_r be the covariance matrix. Most clustering methods depend on the pairwise distances $\|X_i - X_j\|^2$. Now,

$$\|X_i - X_j\|^2 = \sum_{a=1}^d \delta(a)$$

where $\delta_a = (X_i(a) - X_j(a))^2$. This is a sum. As d increases, by the law of large numbers we might expect this sum to converge to a number (assuming the features are not too dependent). Indeed, suppose that X is from P_r and Y is from P_s then

$$\frac{1}{\sqrt{d}} \|X - Y\| \xrightarrow{P} \sqrt{\sigma_r^2 + \sigma_s^2 + \nu_{rs}}$$

where

$$\nu_{rs} = \lim_{d \rightarrow \infty} \frac{1}{d} \sum_{a=1}^d \|\mu_r(a) - \mu_s(a)\|^2$$

and

$$\sigma_r^2 = \lim_{d \rightarrow \infty} \frac{1}{d} \text{trace}(\Sigma_r).$$

Note that $\nu_{rr} = 0$.

Consider two clusters, C_1 and C_2 :

X	Y	$\ X - Y\ $
$X \in C_1$	$Y \in C_1$	$\ X - Y\ = 2\sigma_1^2$
$X \in C_2$	$Y \in C_2$	$\ X - Y\ = 2\sigma_2^2$
$X \in C_1$	$Y \in C_2$	$\ X - Y\ = \sigma_1^2 + \sigma_2^2 + \nu_{12}$

If

$$\sigma_1^2 + \nu_{12} < \sigma_2^2$$

then **every point in cluster 2 is closer to a point in cluster 1 than to other points in cluster 2**. Indeed, if you simulate high dimensional Gaussians, you will see that all the standard clustering methods fail terribly.

What's really going on is that high dimensional data tend to cluster on rings. Pairwise distance methods don't respect rings.

An interesting fix suggested by Sarkar and Ghosh is to use the mean absolute difference distance (MADD) defined by

$$\rho(x, y) = \frac{1}{n-2} \sum_{z \neq x, y} \left| \|x - z\| - \|y - z\| \right|.$$

Suppose that $X \sim P_r$ and $Y \sim P_s$. They show that $\rho(X, Y) \xrightarrow{P} c_{rs}$ where $c_{rs} \geq 0$ and $c_{rs} = 0$ if and only if $\sigma_r^2 = \sigma_s^2$ and $\nu_{br} = \nu_{bs}$ for all b . What this means is that pairwise distance methods only work if $\nu_{rs} > |\sigma_r^2 - \sigma_s^2|$ but MADD works if either $\nu_{rs} \neq 0$ or $\sigma_r \neq \sigma_s$.

Pairwise distances only use information about two moments and they combine this moment information in a particular way. MADD combines the moment information in a different and more effective way. One could also invent other measures that separate mean and variance information or that use higher moment information.

8.2 Variable Selection

If $X \in \mathbb{R}^d$ is high dimensional, then it makes sense to do variable selection before clustering. There are a number of methods for doing this. But, frankly, none are very convincing. This is, in my opinion, an open problem. Here are a couple of possibilities.

Marginal Selection (Screening). In marginal selection, we look for variables that marginally look ‘clustery.’ This idea was used in Chan and Hall (2010) and Wasserman, Azizyan and Singh (2014). We proceed as follows:

Test For Multi-Modality

1. Fix $0 < \alpha < 1$. Let $\tilde{\alpha} = \alpha/(nd)$.
2. For each $1 \leq j \leq d$, compute $T_j = \text{Dip}(F_{nj})$ where F_{nj} is the empirical distribution function of the j^{th} feature and $\text{Dip}(F)$ is defined in (22).
3. Reject the null hypothesis that feature j is not multimodal if $T_j > c_{n,\tilde{\alpha}}$ where $c_{n,\tilde{\alpha}}$ is the critical value for the dip test.

Any test of multimodality may be used. Here we describe the *dip test* (Hartigan and Hartigan, 1985). Let $Z_1, \dots, Z_n \in [0, 1]$ be a sample from a distribution F . We want to test “ $H_0 : F$ is unimodal” versus “ $H_1 : F$ is not unimodal.” Let \mathcal{U} be the set of unimodal

distributions. Hartigan and Hartigan (1985) define

$$\text{Dip}(F) = \inf_{G \in \mathcal{U}} \sup_x |F(x) - G(x)|. \quad (22)$$

If F has a density p we also write $\text{Dip}(F)$ as $\text{Dip}(p)$. Let F_n be the empirical distribution function. The dip statistic is $T_n = \text{Dip}(F_n)$. The dip test rejects H_0 if $T_n > c_{n,\alpha}$ where the critical value $c_{n,\alpha}$ is chosen so that, under H_0 , $\mathbb{P}(T_n > c_{n,\alpha}) \leq \alpha$.²

Since we are conducting multiple tests, we cannot test at a fixed error rate α . Instead, we replace α with $\tilde{\alpha} = \alpha/(nd)$. That is, we test each marginal and we reject H_0 if $T_n > c_{n,\tilde{\alpha}}$. By the union bound, the chance of at least one false rejection of H_0 is at most $d\tilde{\alpha} = \alpha/n$.

There are more refined tests such as the excess mass test given in Chan and Hall (2010), building on work by Muller and Sawitzki (1991). For simplicity, we use the dip test in this paper; a fast implementation of the test is available in R.

Marginal selection can obviously fail. See Figure 28 taken from Wasserman, Azizyan and Singh (2014).

Sparse k -means. Here we discuss the approach in Witten and Tibshirani (2010). Recall that in k -means clustering we choose $C = \{c_1, \dots, c_k\}$ to minimize

$$R_n(C) = \frac{1}{n} \sum_{i=1}^n \|X_i - \Pi_C[X_i]\|^2 = \frac{1}{n} \sum_{i=1}^n \min_{1 \leq j \leq k} \|X_i - c_j\|^2. \quad (23)$$

This is equivalent to minimizing the within sums of squares

$$\sum_{j=1}^k \frac{1}{n_j} \sum_{s,t \in A_j} d^2(X_s, X_t) \quad (24)$$

where A_j is the j^{th} cluster and $d^2(x, y) = \sum_{r=1}^d (x(r) - y(r))^2$ is squared Euclidean distance. Further, this is equivalent to maximizing the between sums of squares

$$B = \frac{1}{n} \sum_{s,t} d^2(X_s, X_t) - \sum_{j=1}^k \frac{1}{n_j} \sum_{s,t \in A_j} d^2(X_s, X_t). \quad (25)$$

Witten and Tibshirani propose replace the Euclidean norm with the weighted norm $d_w^2(x, y) = \sum_{r=1}^d w_r(x(r) - y(r))^2$. Then they propose to maximize

$$B = \frac{1}{n} \sum_{s,t} d_w^2(X_s, X_t) - \sum_{j=1}^k \frac{1}{n_j} \sum_{s,t \in A_j} d_w^2(X_s, X_t) \quad (26)$$

²Specifically, $c_{n,\alpha}$ can be defined by $\sup_{G \in \mathcal{U}} P_G(T_n > c_{n,\alpha}) = \alpha$. In practice, $c_{n,\alpha}$ can be defined by $P_U(T_n > c_{n,\alpha}) = \alpha$ where U is $\text{Unif}(0,1)$. Hartigan and Hartigan (1985) suggest that this suffices asymptotically.

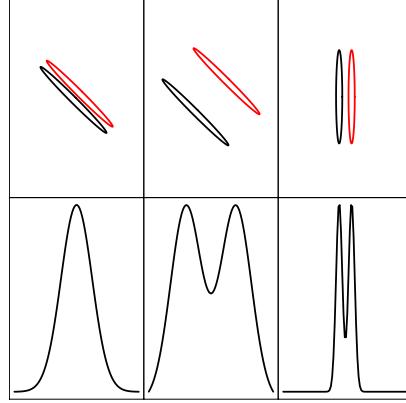


Figure 28: Three examples, each showing two clusters and two features $X(1)$ and $X(2)$. The top plots show the clusters. The bottom plots show the marginal density of $X(1)$. Left: The marginal fails to reveal any clustering structure. This example violates the marginal signature assumption. Middle: The marginal is multimodal and hence correctly identifies $X(1)$ as a relevant feature. This example satisfies the marginal signature assumption. Right: In this case, $X(1)$ is relevant but $X(2)$ is not. Despite the fact that the clusters are close together, the marginal is multimodal and hence correctly identifies $X(1)$ as a relevant feature. This example satisfies the marginal signature assumption. (Figure from Wasserman, Azizyan and Singh, 2014).

over C and w subject to the constraints

$$\|w\|^2 \leq 1, \quad \|w\|_1 \leq s, \quad w_j \geq 0$$

where $w = (w_1, \dots, w_d)$. The optimization is done iteratively by optimizing over C , optimizing over w and repeating. See Figure 29.

The ℓ_1 norm on the weights causes some of the components of w to be 0 which results in variable selection. There is no theory that shows that this method works.

Sparse Alternate Sum Clustering. Arais-Castro and Pu (arXiv:1602.07277) introduced a method called SAS (Sparse Alternate Sum) clustering. It is very simple and intuitively appealing.

Recall that k -means minimizes

$$\sum_j \frac{1}{|C_j|} \sum_{i,j \in C_j} \|X_i - X_j\|^2.$$

Suppose we want a clustering based on a subset of features S such that $|S| = L$. Let $\delta_a(i, j) = (X_i(a) - X_j(a))^2$ be the pairwise distance for the a^{th} feature. Assume that each

1. Input X_1, \dots, X_n and k .
2. Set $w = (w_1, \dots, w_d)$ where $w_1 = \dots = w_d = 1/\sqrt{d}$.
3. Iterate until convergence:
 - (a) Optimize (25) over C holding w fixed. Find c_1, \dots, c_k from the k -means algorithm using distance $d_w(X_i, X_j)$. Let A_j denote the j^{th} cluster.
 - (b) Optimize (25) over w holding c_1, \dots, c_k fixed. The solution is

$$w_r = \frac{s_r}{\sqrt{\sum_{t=1}^d s_t^2}}$$

where

$$s_r = (a_r - \Delta)_+,$$

$$a_r = \left[\frac{1}{n} \sum_{s,t} w_r (X_s(r) - X_t(r))^2 - \sum_{j=1}^k \frac{1}{n_j} \sum_{s,t \in A_j} w_r (X_s(r) - X_t(r))^2 \right]_+$$

and $\Delta = 0$ if $\|w\|_1 < s$ otherwise $\Delta > 0$ is chosen to that $\|w\|_1 = s$.

Figure 29: The Witten-Tibshirani Sparse k -means Method

feature has been standardized so that

$$\sum_{i,j} \delta_a(i,j) = 1$$

for all a . Define $\delta_S(i,j) = \sum_{a \in S} \delta_a(i,j)$. Then we can say that the goal of sparse clustering is to minimize

$$\sum_j \frac{1}{|C_j|} \sum_{i,j \in C_j} \delta_S(i,j)$$

over clusterings and subsets. They propose to minimize by alternating between finding clusters and finding subsets. The former is the usual k -means. The latter is trivial because δ_S decomposes into marginal components. Arias-Castro and Pu also suggest a permutation method for choosing the size of S . Their numerical experiments are very promising. Currently, no theory has been developed for this approach.

8.3 Mosaics

A different idea is to create a partition of features and observations which I like to call a *mosaic*. There are papers that cluster features and observations simultaneously but clear theory is still lacking.

9 Examples

Example 21 Figures 17 and 18 shows some synthetic examples where the clusters are meant to be intuitively clear. In Figure 17 there are two blob-like clusters. Identifying clusters like this is easy. Figure 18 shows four clusters: a blob, two rings and a half ring. Identifying clusters with unusual shapes like this is not quite as easy. To the human eye, these certainly look like clusters. But what makes them clusters?

Example 22 (Gene Clustering) In genomic studies, it is common to measure the expression levels of d genes on n people using microarrays (or gene chips). The data (after much simplification) can be represented as an $n \times d$ matrix X where X_{ij} is the expression level of gene j for subject i . Typically d is much larger than n . For example, we might have $d \approx 5,000$ and $n \approx 50$. Clustering can be done on genes or subjects. To find groups of similar people, regard each row as a data vector so we have n vectors X_1, \dots, X_n each of length d . Clustering can then be used to place the subjects into similar groups.

Example 23 (Curve Clustering) Sometimes the data consist of a set of curves f_1, \dots, f_n and the goal is to cluster similarly shaped clusters together. For example, Figure 30 shows a

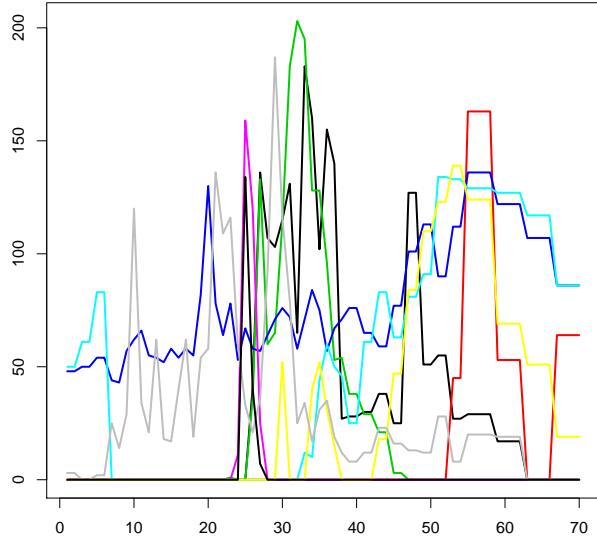


Figure 30: Some curves from a dataset of 472 curves. Each curve is a radar waveform from the Topex/Poseidon satellite.

small sample of curves a from a dataset of 472 curves from Frappart (2003). Each curve is a radar waveform from the Topex/Poseidon satellite which used to map the surface topography of the oceans.³ One question is whether the 472 curves can be put into groups of similar shape.

Example 24 (Supernova Clustering) *Figure 31 shows another example of curve clustering. Briefly, each data point is a light curve, essentially brightness versus time. The top two plots show the light curves for two types of supernovae called “Type Ia” and “other.” The bottom two plots show what happens if we throw away the labels (“Type Ia” and “other”) and apply a clustering algorithm (k -means clustering). We see that the clustering algorithm almost completely recovers the two types of supernovae.*

³See <http://topex-www.jpl.nasa.gov/overview/overview.html>. The data are available at “Working Group on Functional and Operator-based Statistics” a web site run by Frederic Ferraty and Philippe Vieu. The address is <http://www.math.univ-toulouse.fr/staph/npfda/>. See also http://podaac.jpl.nasa.gov/DATA_CATALOG/topexPoseidoninfo.html.

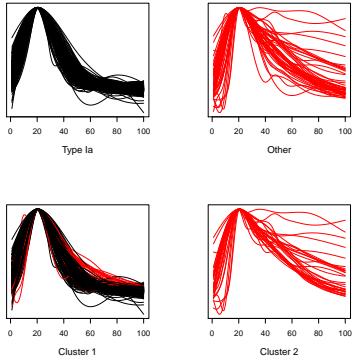


Figure 31: Light curves for supernovae. The top two plots show the light curves for two types of supernovae. The bottom two plots show the results of clustering the curves into two groups, without using knowledge of their labels.

10 Bibliographic Remarks

k -means clustering goes back to Stuart Lloyd who apparently came up with the algorithm in 1957 although he did not publish it until 1982. See [?]. Another key reference is [?]. Similar ideas appear in [?]. The related area of mixture models is discussed at length in McLachlan and Basford (1988). k -means is actually related to principal components analysis; see Ding and He (2004) and Zha, He, Ding, Simon and Gu (2001). The probabilistic behavior of random geometric graphs is discussed in detail in [?].

Undirected Graphical Models

36-708

Contents

1 Marginal Correlation Graphs	2
2 Partial Correlation Graphs	10
3 Conditional Independence Graphs	13
3.1 Gaussian	13
3.2 Multinomials and Log-Linear Models	14
3.3 The Nonparametric Case	16
4 A Deeper Look At Conditional Independence Graphs	26
4.1 Markov Properties	26
4.2 Clique Decomposition	29
4.3 Directed vs. Undirected Graphs	32
4.4 Faithfulness	36

Graphical models are a way of representing the relationships between features (variables). There are two main brands: directed and undirected. We shall focus on undirected graphical models. See Figure 1 for an example of an undirected graph.

Undirected graphs come in different flavors, such as:

1. Marginal Correlation Graphs.
2. Partial Correlation Graphs.
3. Conditional Independence Graphs.

In each case, there are parametric and nonparametric versions.

Let $X_1, \dots, X_n \sim P$ where $X_i = (X_i(1), \dots, X_i(d))^T \in \mathbb{R}^d$. The vertices (nodes) of the graph refer to the d features. Each node of the graph corresponds to one feature. Edges represent relationships between the features. The graph is represented by $G = (V, E)$ where $V = (V_1, \dots, V_d)$ are the vertices and E are the edges. We can regard the edges E as a $d \times d$ matrix where $E(j, k) = 1$ if there is an edge between feature j and feature k and 0 otherwise. Alternatively, you can regard E as a list of pairs where $(j, k) \in E$ if there is an edge between j and k . We write

$$X \amalg Y$$

to mean that X and Y are independent. In other words, $p(x, y) = p(x)p(y)$. We write

$$X \amalg Y | Z$$

to mean that X and Y are independent given Z . In other words, $p(x, y|z) = p(x|z)p(y|z)$.

1 Marginal Correlation Graphs

In a marginal correlation graph (or association graph) we put an edge between V_j and V_k if $|\rho(j, k)| \geq \epsilon$ where $\rho(j, k)$ is some measure of association. Often we use $\epsilon = 0$ in which case there is an edge iff $\rho(j, k) \neq 0$. We also write $\rho(X_j, X_k)$ to mean the same as $\rho(j, k)$.

The parameter $\rho(j, k)$ is required to have the following property:

$$X \amalg Y \text{ implies that } \rho(X, Y) = 0.$$

In general, the reverse may not be true. We will say that ρ is *strong* if

$$X \amalg Y \text{ if and only if } \rho(X, Y) = 0.$$

We would like ρ to have several properties: easy to compute, robust to outliers and there is some way to calculate a confidence interval for the parameter. Here is a summary of the association measures we will consider:

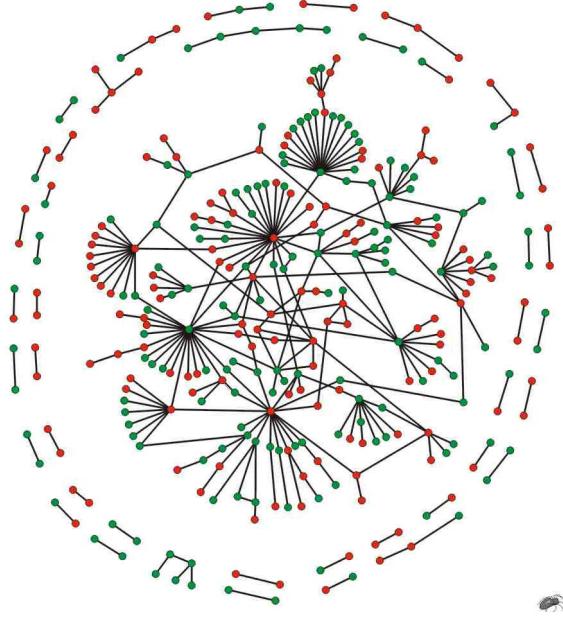


Figure 1: A Protein network. From: Maslov and Sneppen (2002). Specificity and Stability in Topology of Protein Networks. Science, 296, 910-913.

	Strong	Robust	Fast	Confidence Interval
Pearson	✗	✗	✓	✓
Kendall	✗	✓	✓	✓
Dcorr	✓	✗	✗	sort of
τ^*	✓	✓	✗	✓

Pearson Correlation. A common choice of ρ is the Pearson correlation. For two variables X and Y te Pearson correlation is

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \quad (1)$$

. The sample estimate is

$$r(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{s_X s_Y}.$$

When dealing with d feaures $X(1), \dots, X(d)$, we write $\rho(j, k) \equiv \rho(X(j), X(k))$. The sample correlation is denoted by r_{jk} .

To test $H_0 : \rho(j, k) = 0$ versus $H_1 : \rho(j, k) \neq 0$ we can use an asymptotic test or an exact test.

The asymptotic test works like this. Define

$$Z_{jk} = \frac{1}{2} \log \left(\frac{1 + r_{jk}}{1 - r_{jk}} \right).$$

Fisher proved that

$$Z_{jk} \approx N \left(\theta_{jk}, \frac{1}{n-3} \right)$$

where

$$\theta_{jk} = \frac{1}{2} \log \left(\frac{1 + \rho_{jk}}{1 - \rho_{jk}} \right).$$

We reject H_0 if $|Z_{jk}| > z_{\alpha/2}/\sqrt{n-3}$. In fact, to control for multiple testing, we should reject when $|Z_{jk}| > z_{\alpha/(2m)}/\sqrt{n-3}$ where $m = \binom{d}{2}$. The confidence interval is $C_n = [a, b]$ where $a = \exp(Z_{jk} - z_{\alpha/2}/\sqrt{n-3})$ and $b = \exp(Z_{jk} + z_{\alpha/2}/\sqrt{n-3})$. A simultaneous confidence set for all the correlations can be obtained using the high dimensional bootstrap which we describe later.

An exact test can be obtained by using a permutation test. Permute one of the variables and recompute the correlation. Repeat B times to get $r_{jk}^1, \dots, r_{jk}^B$. The p-value is

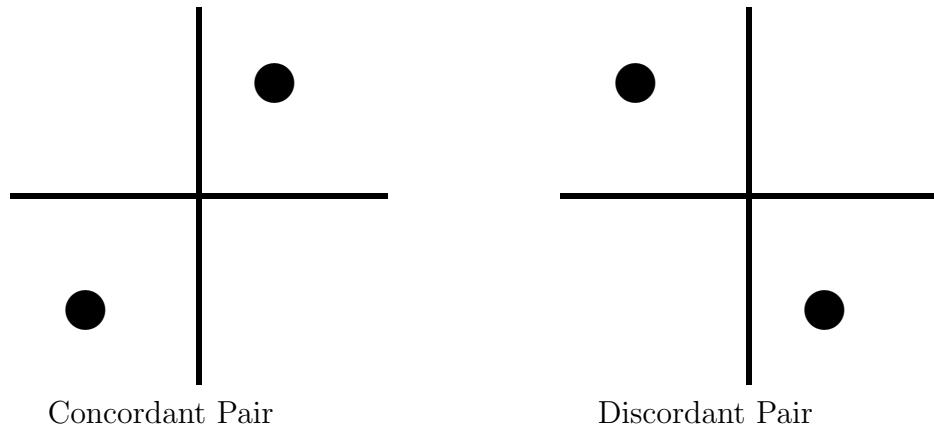
$$p = \frac{1}{B} \sum_s I(|r_{jk}^s| \geq |r_{jk}|).$$

Reject if $p \leq \alpha/m$.

Kendall's τ . The Pearson correlation is not very robust to outliers. A more robust measure of association is Kendall's tau defined by

$$\tau(X, Y) = \mathbb{E} \left[\text{sign}[(X_1 - X_2)(Y_1 - Y_2)] \right].$$

Kendall's τ can be interpreted as: probability(concordant) - probability(disconcordant). See this plot:



τ can be estimated by

$$\widehat{\tau}(X, Y) = \frac{1}{\binom{n}{2}} \sum_{s \neq t} \left[\text{sign}[(X_s - X_t)(Y_s - Y_t)] \right].$$

A statistic of this form is called a U -statistic. Under H_0 , $\widehat{\tau}_{jk} \approx N(0, 4/(9n))$ so we reject when $\sqrt{9n/4}|\widehat{\tau}_{jk}| > z_{\alpha/2m}$. Alternatively, use the permutation test.

Distance Correlation. There are various nonparametric measures of association. The most common are the distance correlation and the RKHS correlation. The squared *distance covariance* between two random vectors X and Y is defined by (Szekely et al 2007)

$$\gamma^2(X, Y) = \text{Cov}(\|X - X'\|, \|Y - Y'\|) - 2\text{Cov}(\|X - X'\|, \|Y - Y''\|) \quad (2)$$

where (X, Y) , (X', Y') and (X'', Y'') are independent pairs. We can write this as

$$\gamma^2(X, Y) = \frac{1}{4} \mathbb{E}[b(X_1, X_2, X_3, X_4)b(Y_1, Y_2, Y_3, Y_4)]$$

where

$$b(z_1, z_2, z_3, z_4) = |z_1 - z_2| + |z_3 - z_4| - |z_1 - z_3| - |z_2 - z_4|$$

The distance correlation is

$$\rho^2(X, Y) = \frac{\gamma^2(X, Y)}{\sqrt{\gamma^2(X, X)\gamma^2(Y, Y)}}.$$

It can be shown that

$$\gamma^2(X, Y) = \frac{1}{c_1 c_2} \int \frac{|\phi_{X,Y}(s, t) - \phi_X(s)\phi_Y(t)|^2}{\|s\|^{1+d}\|t\|^{1+d}} ds dt \quad (3)$$

where c_1, c_2 are constants and ϕ denotes the characteristic function. Another expression (Lyons 2013) for γ is

$$\gamma^2(X, Y) = \mathbb{E}[\delta(X, X')\delta(Y, Y')]$$

where

$$\delta(X, X') = d(X, X') - 2 \int d(X, u)dP(u) + \int \int d(u, v)dP(u)dP(v)$$

and $d(x, y) = \|x - y\|$. In fact, other metrics d can be used.

Lemma 1 *We have that $0 \leq \rho(X, Y) \leq 1$ and $\rho(X, Y) = 0$ if and only if $X \perp\!\!\!\perp Y$.*

An estimate of γ is

$$\hat{\gamma}^2(X, Y) = \frac{1}{n^2} \sum_{j,k} A_{jk} B_{jk}$$

where

$$A_{jk} = a_{jk} - a_{j\cdot} - a_{\cdot k} + a_{\cdot\cdot}, \quad B_{jk} = b_{jk} - b_{j\cdot} - b_{\cdot k} + b_{\cdot\cdot}$$

Here, $a_{jk} = \|X_j - X_k\|$ and $a_{j\cdot}, a_{\cdot k}, a_{\cdot\cdot}$ are the row, column and grand means of the matrix $\{a_{jk}\}$. The limiting distribution of $\hat{\gamma}^2(X, Y)$ is complicated. But we can easily test $H_0 : \gamma(X, Y) = 0$ using a permutation test.

Another nonparametric measure of independence based on RKHS is

$$\begin{aligned} \gamma^2(X, Y) &= \mathbb{E}[K_h(X, X')K_h(Y, Y')] + \mathbb{E}[K_h(X, X')]\mathbb{E}[K_h(Y, Y')] \\ &\quad - 2\mathbb{E}\left[\int K_h(X, u)dP(u) \int K_h(Y, v)dP(v)\right] \end{aligned}$$

for a kernel K_h . See Gretton et al (2008).

To apply any of these methods to graphs, we need to test all $\binom{d}{2}$ correlations.

The Bergsma-Dassios τ^* Correlation. Bergsma and Dassios (2014) extended Kendall's τ into a strong correlation. The definition is

$$\tau^*(X, Y) = \mathbb{E}[a(X_1, X_2, X_3, X_4)a(Y_1, Y_2, Y_3, Y_4)] \tag{4}$$

where

$$a(z_1, z_2, z_3, z_4) = \text{sign}(|z_1 - z_2| + |z_3 - z_4| - |z_1 - z_3 - |z_2 - z_4||).$$

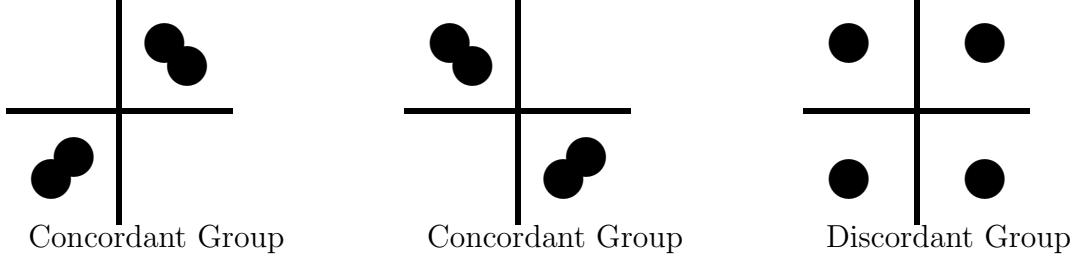
Lemma 2 $\tau^*(X, Y) \geq 0$. Further, $\tau^*(X, Y) = 0$ if and only if $X \perp\!\!\!\perp Y$.

An estimate of τ^* is

$$\hat{\tau}^* = \frac{1}{\binom{n}{4}} \sum a(X_i, X_j, X_k, X_\ell)a(Y_i, Y_j, Y_k, Y_\ell) \tag{5}$$

where the sum is over all distinct quadruples.

The τ^* parameter can also be given an interpretation in terms of concordant and discordant points if we define them as follows:



Then

$$\tau^* = \frac{2P(\text{concordant}) - P(\text{discordant})}{3}.$$

This statistic is related to the distance covariance as follows:

$$\begin{aligned}\tau^*(X, Y) &= \mathbb{E}[a(X_1, X_2, X_3, X_4)a(Y_1, Y_2, Y_3, Y_4)] \\ \gamma^2(X, Y) &= \frac{1}{4}\mathbb{E}[b(X_1, X_2, X_3, X_4)b(Y_1, Y_2, Y_3, Y_4)]\end{aligned}$$

where

$$b(z_1, z_2, z_3, z_4) = |z_1 - z_2| + |z_3 - z_4| - |z_1 - z_3| - |z_2 - z_4|$$

and $a(z_1, z_2, z_3, z_4) = \text{sign}(b(z_1, z_2, z_3, z_4))$.

To test $H_0 : X \perp\!\!\!\perp Y$ we use a permutation test. Recently Dhar, Dassios and Bergsma (2016) showed that $\hat{\tau}^*$ has good power and is quite robust.

Confidence Intervals. Constructing confidence intervals for γ is not easy. The problem is that the statistics have different limiting distributions depending on whether the null $H_0 : X \perp\!\!\!\perp Y$ is true or not. For example, if H_0 is true then

$$n\hat{\gamma}^2 \rightsquigarrow \sum_{j=1}^{\infty} \lambda_j[(Z_j + a_j)^2 - 1]$$

where $Z_1, Z_2, \dots, N(0, 1)$ and $\{\lambda_j, a_j\}_1^{\infty}$ are (unknown) constants. A similar result holds for $\hat{\gamma}$. This is called a Gaussian chaos. On the other hand, when H_0 is false, the limiting distribution is different.

Since the limiting distribution varies, we cannot really use it to construct a confidence interval. One way to solve this problem is to use blocking. Instead of using a U-statistics based on all subsets of size 4, we can break the dataset into non-overlapping blocks of size 4. We construct $Q = a(X_i, X_j, X_k, X_\ell)a(Y_i, Y_j, Y_k, Y_\ell)$ on each block. Then we define

$$g = \frac{1}{m} \sum_j Q_j$$

where $m = n/4$ is the number of blocks. Since this is an average, it will have a limiting Normal distribution. We can then use the Normal approximation or the bootstrap to get confidence intervals. However, g uses less information than $\hat{\gamma}$ so we will get larger confidence intervals than necessary.

For τ^* the situation is better. Note that $\hat{\tau}^*$ is a U-statistic of order 4. That is

$$\hat{\tau}^* = \hat{\tau}^* = \frac{1}{\binom{n}{4}} \sum K(Z_i, Z_j, Z_k, Z_\ell)$$

where the sum is over all distinct quadruples and $Z_i = (X_i, Y_i)$. Also, $-1 \leq K \leq 1$. By Hoeffding's inequality for U -statsitics of order b we have

$$\mathbb{P}(|\hat{\tau}^* - \tau^*| > t) \leq 2e^{-2(n/b)t^2/r^2}$$

where r is the range of K . In our case, $r = 2$ and $b = 4$ so

$$\mathbb{P}(|\hat{\tau}^* - \tau^*| > t) \leq 2e^{-nt^2/8}.$$

If we set $t_n = \sqrt{(8/n) \log(2/\alpha)}$ then $C_n = \hat{\tau}^* \pm t_n$ is a $1 - \alpha$ confidence interval. However, it may not be shortest possible confidence interval. Find the shortest valid confidence interval is an open question.

Example. Figure 2 shows some Pearson graphs. These are: two Markov chains, a hub, four clusters, and a band. Technically, these should be best discovered using conditional independence graphs (discussed later). But correlation graphs are easy to estimate and often reveal the salient structure.

Figure 3 shows a graph from highly non-Normal data. The data have the structure of two Markov chains. I used the distance correlation on all pairs with permutation tests. Nice!

High Dimensional Bootstrap For Pearson Correlations We can also get simultaneous confidence intervals for many Pearson correlations. This is especially important if we want to put an edge when $|\rho(j, k)| \geq \epsilon$. If we have a confidence interval C then we can put an edge whenever $[-\epsilon, \epsilon] \cap C = \emptyset$.

The easiest way to get simultaneous confidence intervals is to use the bootstrap. Let R be the $d \times d$ matrix of true correlations and let \hat{R} be the $d \times d$ matrix of sample correlations. (Actually, it is probably better to use the Fisher transformed correlations.) Let X_1^*, \dots, X_n^* denote a bootstrap sample and let \hat{R}^* be the $d \times d$ matrix of correlations from the bootstrap sample. After taking B bootstrap samples we have $\hat{R}_1^*, \dots, \hat{R}_B^*$. Let $\delta_j = \sqrt{n} \max_{s,t} |\hat{R}_j^*(s, t) - \hat{R}(s, t)|$ and define

$$\hat{F}_n(w) = \frac{1}{B} \sum_{j=1}^B I(\delta_j \leq w)$$

which approximates

$$F_n(w) = \mathbb{P}(\sqrt{n} \max_{s,t} |\hat{R}(s, t) - R(s, t)| \leq w).$$

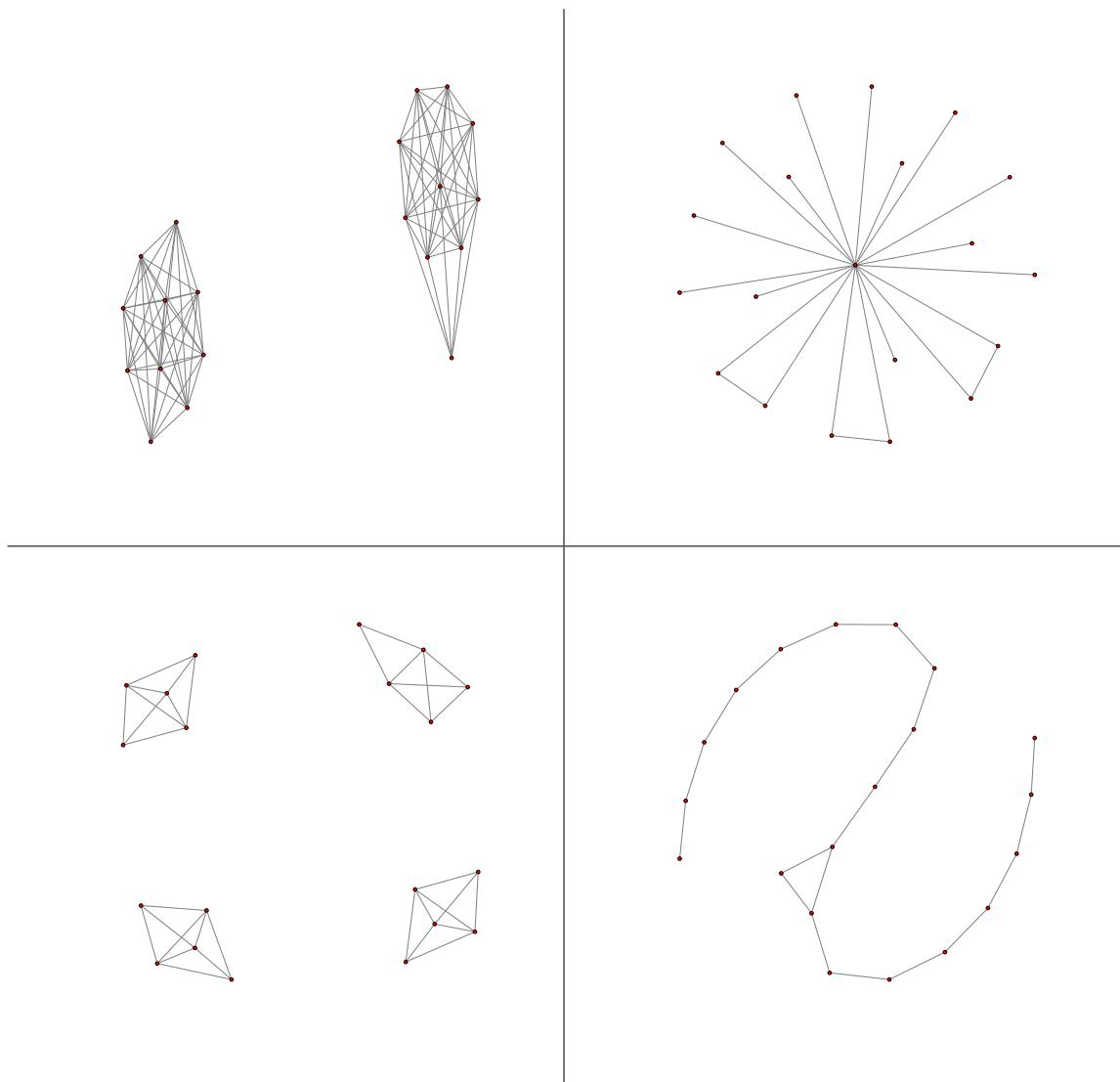


Figure 2: Pearson correlation graphs. Top left: two Markov-chains. Top right: a Hub. Bottom left: 4 clusters. Bottom right: banded.

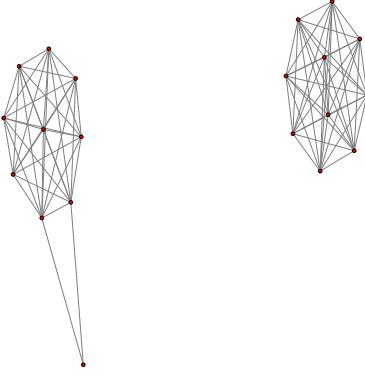


Figure 3: Graph based on nonparametric distance correlation. Two Markov chains.

Let $w_\alpha = \widehat{F}_n^{-1}(\alpha)$. Finally, we set

$$C_{st} = \left[\widehat{R}(s, t) - \frac{w_\alpha}{\sqrt{n}}, \quad \widehat{R}(s, t) + \frac{w_\alpha}{\sqrt{n}} \right].$$

Theorem 3 Suppose that $d = o(e^{n^{1/6}})$. Then

$$\mathbb{P}(R(s, t) \in C_{st} \text{ for all } s, t) \rightarrow 1 - \alpha$$

as $n \rightarrow \infty$.

2 Partial Correlation Graphs

Let $X, Y \in \mathbb{R}$ and Z be a random vector. The partial correlation between X and Y , given Z , is a measure of association between X and Y after removing the effect of Z .

Specifically, $\rho(X, Y|Z)$ is the correlation between ϵ_X and ϵ_Y where

$$\epsilon_X = X - \Pi_Z X, \quad \epsilon_Y = Y - \Pi_Z Y.$$

Here, $\Pi_Z X$ is the projection of X onto the linear space spanned by Z . That is $\Pi_Z X = \beta^T X$ where β minimizes $\mathbb{E}[Y - \beta^T X]^2$. In other words, $\Pi_Z X$ is the linear regression of X on Z . Similarly, for $\Pi_Z Y$. We'll give an explicit formula for the partial correlation shortly.

Now let's go back to graphs. Let $X = (X(1), \dots, X(d))$ and let ρ_{jk} denote the partial correlation between $X(j)$ and $X(k)$ given all the other variables. Let $R = \{\rho_{jk}\}$ be the $d \times d$ matrix of partial correlations.

Lemma 4 *The matrix R is given by $R(j, k) = -\Omega_{jk}/\sqrt{\Omega_{jj}\Omega_{kk}}$ where $\Omega = \Sigma^{-1}$ and Σ is the covariance matrix of X .*

The partial correlation graph G has an edge between j and k when $\rho_{jk} \neq 0$.

In the low-dimensional setting, we can estimate R as follows. Let S_n be the sample covariance. Let $\widehat{\Omega} = S_n^{-1}$ and define $\widehat{R}(j, k) = -\widehat{\Omega}_{jk}/\sqrt{\widehat{\Omega}_{jj}\widehat{\Omega}_{kk}}$. The easiest way to construct the graph is to use get simultaneous confidence intervals C_{jk} using the bootstrap. Then we put an edge if $0 \notin C_{jk}$.

There is also a Normal approximation similar to correlations. Define

$$Z_{jk} = \frac{1}{2} \log \left(\frac{1 + r_{jk}}{1 - r_{jk}} \right)$$

where $r_{jk} = \widehat{R}(j, k)$. Then

$$Z_{jk} \approx N \left(\theta_{jk}, \frac{1}{n - g - 3} \right)$$

where $g = d - 2$ and

$$\theta_{jk} = \frac{1}{2} \log \left(\frac{1 + \rho_{jk}}{1 - \rho_{jk}} \right).$$

We reject H_0 if $|Z_{jk}| > z_{\alpha/(2m)}/\sqrt{n - g - 3}$.

In high dimensions, this won't work since S_n is not invertible. In fact,

$$\text{Var}(\widehat{R}(j, k)) \approx \frac{1}{n - d}$$

which shows that we cannot reliably estimate the partial correlation when d is large. You can do three things:

1. Compute a correlation graph instead. This is easy, works well, and often reveals similar structure that is in the partial correlation graph.
2. Shrinkage: let $\widehat{\Omega} = [(1 - \epsilon)S_n + \epsilon D]^{-1}$ where $0 \leq \epsilon \leq 1$ and D is a diagonal matrix with $D_{jj} = S_{jj}$. Then we use the bootstrap to test the entries of the matrix. Based on calculations in Schafer and Strimmer (2005) and Ledoit and Wolf (2004), a good choice of ϵ is

$$\epsilon = \frac{\sum_{j \neq k} \widehat{\text{Var}}(s_{jk})}{\sum_{j \neq k} s_{jk}^2}$$

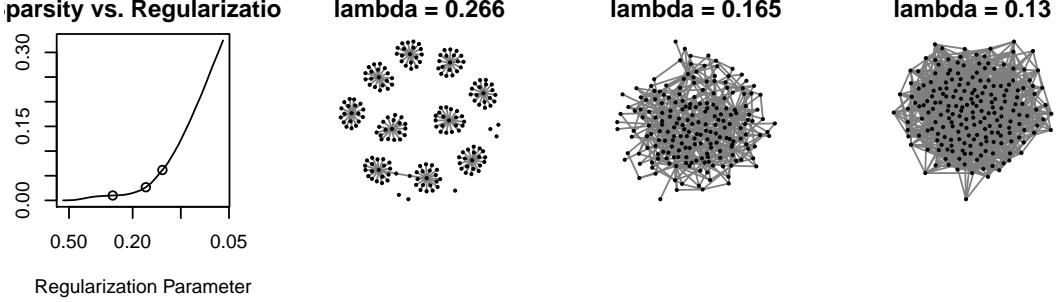


Figure 4: Graphical Lasso for a hub graph.

where

$$\widehat{\text{Var}}(s_{jk}) = \frac{n}{(n-1)^3} \sum_{i=1}^n (s_{ijk} - \bar{w}_{jk})^2,$$

$w_{ijk} = (X_i(j) - \bar{X}(j))(X_i(k) - \bar{X}(k))$ and $\bar{w}_{jk} = n^{-1} \sum_i w_{ijk}$. This choice is based on minimizing the estimated risk.

3. Use the graphical lasso (described below). Warning! The reliability of the graphical lasso depends on lots of non-trivial, uncheckable assumptions.

For the graphical lasso we proceed as follows. We assume that $X_i \sim N(\mu, \Sigma)$. Then we estimate Σ (and hence $\Omega = \Sigma^{-1}$) using the penalized log-likelihood,

$$\widehat{\Omega} = \arg \max_{\Omega \succ 0} \left[\ell(\Omega) - \lambda \sum_{j \neq k} |\omega_{jk}| \right]$$

where the log-likelihood (after maximizing over μ) is

$$\ell(\Omega) = \frac{n}{2} \log |\Omega| - \frac{n}{2} \text{tr}(\Omega S_n) - \frac{nd}{2} \log(2\pi). \quad (6)$$

Node-wise regression. A related but different approach due to Meinshausen and Bühlmann (2006). The idea is to regress each variable on all the others using the lasso.

Example: Figure 4 shows a hub graph. The graphs were estimated by Meinshausen and Bühlmann (2006) method using the R package `huge`.

How To Choose λ . If we use the graphical lasso, how do we choose λ ? One idea is to use cross-validation based on the Normal log-likelihood. In this case we fit the model on part of the data and evaluate the log-likelihood on the held-out data. This is not very reliable since it depends heavily on the Normality assumption. Currently, I do not think there is a rigorously

justified, robust method for choosing λ . Perhaps the best strategy is to plot the graph for many values of λ .

More Robust Approach. Here is a more robust method. For each pair of variables, regress them on all the other variables (using your favorite regression method). Now compute the Kendal correlation on the residuals. This seems like a good idea but I have not seen anyone try it.

Nonparametric Partial Correlation Graphs. There are various ways to create a nonparametric partial correlation. Let us write

$$\begin{aligned} X &= g(Z) + \epsilon_X \\ Y &= h(Z) + \epsilon_Y. \end{aligned}$$

Thus, $\epsilon_X = X - g(Z)$ and $\epsilon_Y = Y - h(Z)$ where $g(z) = \mathbb{E}[X|Z = z]$ and $h(z) = \mathbb{E}[Y|Z = z]$. Now define

$$\rho(X, Y|Z) = \rho(\epsilon_X, \epsilon_Y).$$

We can estimate ρ by using nonparametric regression to estimate $g(z)$ and $h(z)$. Then we take the correlation between the residuals $\hat{\epsilon}_{X,i} = X_i - \hat{g}(X_i)$ and $\hat{\epsilon}_{Y,i} = Y_i - \hat{h}(Y_i)$. When Z is high-dimensional, we can use SpAM to estimate g and h .

3 Conditional Independence Graphs

The strongest type of undirected graph is a *conditional independence graph*. In this case, we omit the edge between j and k if $X(j)$ is independent of $X(k)$ given the rest of the variables. We write this as

$$X(j) \amalg X(k) \mid \text{rest}. \quad (7)$$

Conditional independence graphs are the most informative undirected graphs but they are also the hardest to estimate.

3.1 Gaussian

In the special case of Normality, they are equivalent to partial correlation graphs.

Theorem 5 Suppose that $X = (X(1), \dots, X(d)) \sim N(\mu, \Sigma)$. Let $\Omega = \Sigma^{-1}$. Then $X(j)$ is independent of $X(k)$ given the rest, if and only if $\Omega_{jk} = 0$.

So, in the Normal case, we are back to partial correlations.

3.2 Multinomials and Log-Linear Models

When all the variables are discrete, the joint distribution is multinomial. It is convenient to reparameterize the multinomial in a form known as a *log-linear model*.

Let's start with a simple example. Suppose $X = (X(1), X(2))$ and that each variable is binary. Let

$$p(x_1, x_2) = \mathbb{P}(X_1 = x_1, X_2 = x_2).$$

So, for example, $p(0, 1) = \mathbb{P}(X_1 = 0, X_2 = 1)$. There are four unknown parameters: $p(0, 0), p(0, 1), p(1, 0), p(1, 1)$. Actually, these have to add up to 1, so there are really only three free parameters.

We can now write

$$\log p(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2.$$

This is the log-linear representation of the multinomial. The parameters $\beta = (\beta_0, \beta_1, \beta_2, \beta_{12})$ are functions of $p(0, 0), p(0, 1), p(1, 0), p(1, 1)$. Conversely, $\beta = (\beta_0, \beta_1, \beta_2, \beta_{12})$ are functions of $p(0, 0), p(0, 1), p(1, 0), p(1, 1)$. In fact we can solve and get:

$$\beta_0 = \log p(0, 0), \quad \beta_1 = \log \left(\frac{p(1, 0)}{p(0, 0)} \right), \quad \beta_2 = \log \left(\frac{p(0, 1)}{p(0, 0)} \right), \quad \beta_{12} = \log \left(\frac{p(1, 1)p(0, 0)}{p(0, 1)p(1, 0)} \right).$$

So why should be bother writing the model this way? The answer is:

Lemma 6 *In the above model, $X_1 \perp\!\!\!\perp X_2$ if and only if $\beta_{12} = 0$.*

The log-linear representation converts statements about independence and conditional independence into statements about parameters being 0.

Now suppose that $X = (X(1), \dots, X(d))$. Let's continue to assume that each variable is binary. The log-linear representation is:

$$\log p(x_1, \dots, x_d) = \beta_0 + \sum_j \beta_j x_j + \sum_{j < k} \beta_{jk} x_j x_k + \dots + \beta_{12\dots d} x_1 \dots x_d.$$

Theorem 7 *We have that $X(j) \perp\!\!\!\perp X(k) | \text{rest}$ if and only if every $\beta_A = 0$ if $(j, k) \in A$.*

Here is an example. Suppose that $d = 3$ and suppose that

$$\log p(x_1, x_2, x_3) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3.$$

In this model, $\beta_{23} = \beta_{123} = 0$. We conclude that $X(2) \perp\!\!\!\perp X(3) | X(1)$. Hence, we can omit the edge between $X(2)$ and $X(3)$.

Log-linear models thus make a nice connection between conditional independence graphs and parameters. There is a simple one-line command in R for fitting log-linear models. The function gives the parameter estimates as well as tests that each parameter is 0.

When the variables are not binary, the model is a bit more complicated. Each variable is now represented by a vector of parameters rather than one parameter. But conceptually, it is the same. Suppose $X_j \in \{0, 1, \dots, m - 1\}$, for $j \in V$, with $V = \{1, \dots, d\}$; thus each of the d variables takes one of m possible values.

Definition 8 Let $X = (X_1, \dots, X_d)$ be a discrete random vector with probability function $p(x) = \mathbb{P}(X = x) = \mathbb{P}(X_1 = x_1, \dots, X_d = x_d)$ where $x = (x_1, \dots, x_d)$. The log-linear representation of $p(x)$ is

$$\log p(x) = \sum_{A \subset V} \psi_A(x_A) \tag{8}$$

with the constraints that ψ_Φ is a constant, and if $j \in A$ and $x_j = 0$ then $\psi_A(x_A) = 0$.

The formula in (8) is called the *log-linear expansion* of $p(x)$. Each $\psi_A(x_A)$ may depend on some unknown parameters θ_A . Note that the total number of parameters satisfies $\sum_{j=1}^d \binom{d}{j} (m-1)^j = m^d$, however one of the parameters is the normalizing constant, and is determined by the constraint that the sum of the probabilities is one. Thus, there are $m^d - 1$ free parameters, and this is a minimal exponential parameterization of the multinomial. Let $\theta = (\theta_A : A \subset V)$ be the set of all these parameters. We will write $p(x) = p(x; \theta)$ when we want to emphasize the dependence on the unknown parameters θ .

The next theorem provides an easy way to read out conditional independence in a log-linear model.

Theorem 9 Let (X_A, X_B, X_C) be a partition of $X = (X_1, \dots, X_d)$. Then $X_B \perp\!\!\!\perp X_C | X_A$ if and only if all the ψ -terms in the log-linear expansion that have at least one coordinate in B and one coordinate in C are zero.

Proof. From the definition of conditional independence, we know that $X_B \perp\!\!\!\perp X_C | X_A$ if and only if $p(x_A, x_B, x_C) = f(x_A, x_B)g(x_A, x_C)$ for some functions f and g .

Suppose that ψ_t is 0 whenever t has coordinates in B and C . Hence, ψ_t is 0 if $t \not\subseteq A \cup B$ or $t \not\subseteq A \cup C$. Therefore

$$\log p(x) = \sum_{t \subset A \cup B} \psi_t(x_t) + \sum_{t \subset A \cup C} \psi_t(x_t) - \sum_{t \subset A} \psi_t(x_t). \quad (9)$$

Exponentiating, we see that the joint density is of the form $f(x_A, x_B)g(x_A, x_C)$. Therefore $X_B \perp\!\!\!\perp X_C | X_A$. The reverse follows by reversing the argument. \square

A *graphical log-linear model* with respect to a graph G is a log-linear model for which the parameters ψ_A satisfy $\psi_A(x_A) \neq 0$ if and only if A is a clique of G . Thus, a graphical log-linear model has potential functions on each clique, both maximal and non-maximal, with the restriction that $\psi_A(x_A) = 0$ in case $x_j = 0$ for any $j \in A$. In a *hierarchical log-linear model*, if $\psi_A(x_A) = 0$ then $\psi_B(x_B) = 0$ whenever $A \subset B$. Thus, the parameters in a hierarchical model are nested, in the sense that if a parameter is identically zero for some subset of variables, the parameter for supersets of those variables must also be zero. Every graphical log-linear model is hierarchical, but a hierarchical model need not be graphical; Such a relationship is shown in Figure 5 and is characterized by the next lemma.

Lemma 10 *A graphical log-linear model is hierarchical but the reverse need not be true.*

Proof. We assume there exists a model that is graphical but not hierarchical. There must exist two sets A and B , such that $A \subset B$ with $\psi_A(x_A) = 0$ and $\psi_B(x_B) \neq 0$. Since the model is graphical, $\psi_B(x_B) \neq 0$ implies that B is a clique. We then know that A must also be a clique due to $A \subset B$, which implies that $\psi_A(x_A) \neq 0$. A contradiction.

To see that a hierarchical model does not have to be graphical. We consider the following example. Let

$$\log p(x) = \psi_\Phi + \sum_{i=1}^3 \psi_i(x_i) + \sum_{1 \leq j < k \leq 3} \psi_{jk}(x_{jk}). \quad (10)$$

This model is hierarchical but not graphical. The graph corresponding to this model is a complete graph with three nodes X_1, X_2, X_3 . It is not graphical since $\psi_{123}(x) = 0$, which is contradict with the fact that the graph is complete. \square

3.3 The Nonparametric Case

In real life, nothing has a Normal distribution. What should we do? We could just use a correlation graph or partial correlation graph. That's what I recommend. But if you really want a nonparametric conditional independence graph, there are some possible approaches.

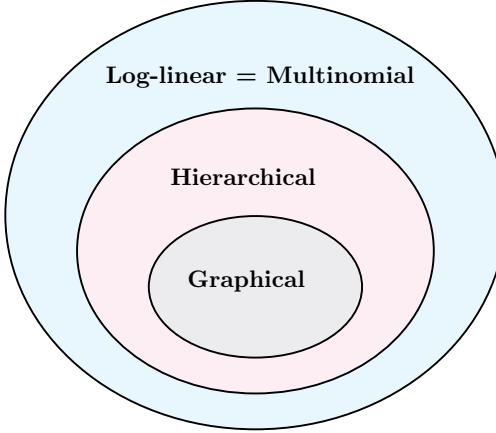


Figure 5: Every graphical log-linear model is hierarchical but the reverse may not be true.

Conditional cdf Method. Let X and Y be real and let Z be a random vector. Define

$$U = F(X|Z), \quad V = G(Y|Z)$$

where

$$F(x|z) = \mathbb{P}(X \leq x|Z = z), \quad G(y|z) = \mathbb{P}(Y \leq y|Z = z).$$

Lemma 11 *If $X \perp\!\!\!\perp Y | Z$ then $U \perp\!\!\!\perp V$.*

If we knew F and G , we could compute $U_i = F(X_i|Z_i)$ and $V_i = G(Y_i|Z_i)$ and then test for independence between U and V .

In practice, we estimate U_i and V_i using smoothing:

$$\widehat{U}_i = \widehat{F}(X_i|Z_i), \quad \widehat{V}_i = \widehat{G}(Y_i|Z_i)$$

where

$$\begin{aligned} \widehat{F}(x|z) &= \frac{\sum_{s=1}^n K(\|z - Z_s\|/h) I(X_s \leq x)}{\sum_{s=1}^n K(\|z - Z_s\|/h)} \\ \widehat{G}(y|z) &= \frac{\sum_{s=1}^n K(\|z - Z_s\|/b) I(Y_s \leq y)}{\sum_{s=1}^n K(\|z - Z_s\|/b)}. \end{aligned}$$

We have the following result from Bergsma (2011).

Theorem 12 (Bergsma) *Let θ be the Pearson or Kendall measure of association between U and V . Let $\widetilde{\theta}$ be the sample version based on (U_i, V_i) , $i = 1, \dots, n$. Let $\widehat{\theta}$ be the sample version based on $(\widehat{U}_i, \widehat{V}_i)$, $i = 1, \dots, n$. Suppose that*

$$n^{1/2}(\widetilde{\theta} - \theta) = O_P(1), \quad n^{\beta_1}(\widehat{F}(x|z) - F(x|z)) = O_P(1), \quad n^{\beta_2}(\widehat{F}(y|z) - F(y|z)) = O_P(1).$$

Then

$$\sqrt{n}(\hat{\theta} - \theta) = \sqrt{n}(\tilde{\theta} - \theta) + O_P(n^{-\gamma})$$

where $\gamma = \min\{\beta_1, \beta_2\}$.

This means that, asymptotically, we can treat (\hat{U}_i, \hat{V}_i) as if they were (U_i, V_i) . Of course, for graphs, the whole procedure needs to be repeated for each pair of variables.

There are some caveats. First, we are essentially doing high dimensional regression. In high dimensions, the convergence will be very slow. Second, we have to choose the bandwidths h and b . It is not obvious how to do this in practice.

Challenge: Can you think of a way to do sparse estimation of $F(x|z)$ and $F(y|z)$?

Nonparanormal. Another approach is to use a Gaussian copula, also known as a *Nonparanormal* (Liu, Lafferty and Wasserman 2009). Recall that, in high dimensional nonparametric regression, we replaced the linear model $Y = \sum_j \beta_j X_j + \epsilon$ with the *sparse additive model*:

$$Y = \sum_j f_j(X_j) + \epsilon \quad \text{where most } \|f_j\| = 0.$$

We can take a similar strategy for graphs.

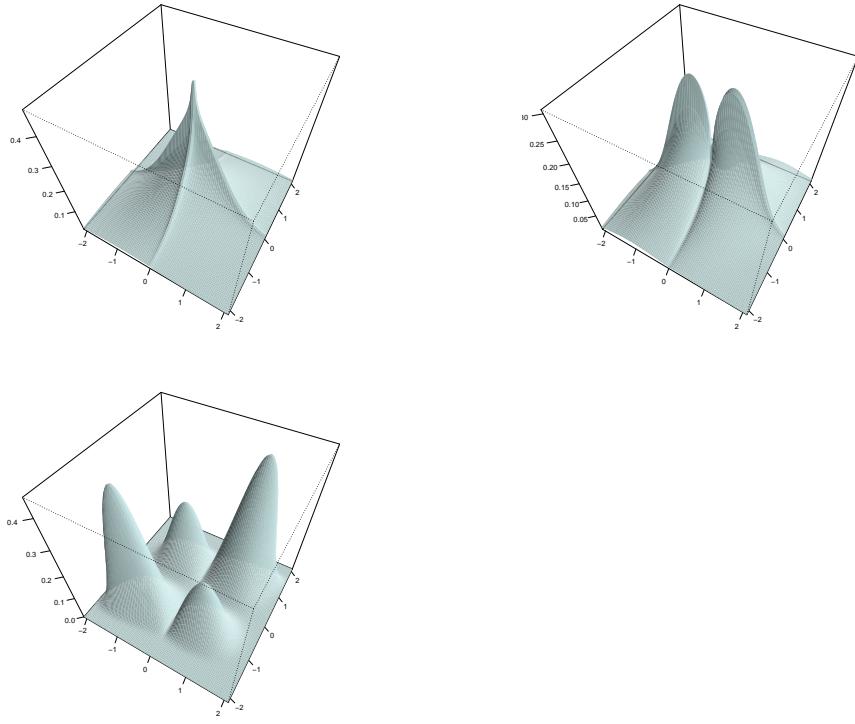
Assumptions	Dimension	Regression	Graphical Models
parametric	low high	linear model lasso	multivariate normal graphical lasso
nonparametric	low high	additive model sparse additive model	nonparanormal ℓ_1 -regularized nonparanormal

One idea is to do node-wise regression using SpAM (see Voorman, Shojaie and Witten 2007). An alternative is as follows. Let $f(X) = (f_1(X_1), \dots, f_p(X_p))$. Assume that $f(X) \sim N(\mu, \Sigma)$. Write $X \sim \text{NPN}(\mu, \Sigma, f)$. If each f_j is monotone then this is just a [Gaussian copula](#), that is,

$$F(x_1, \dots, x_p) = \Phi_{\mu, \Sigma} \left(\Phi^{-1}(F_1(x_1)), \dots, \Phi^{-1}(F_p(x_p)) \right).$$

Lemma 13 $X_j \amalg X_k | \text{rest} \quad \text{iff} \quad \Sigma_{jk}^{-1} = 0$ where $\Sigma = \text{cov}(f(X))$, $f(x) = (f_1(x_1), \dots, f_d(x_d))$ and $f_j(x_j) = \Phi^{-1}(F_j(x_j))$.

The marginal means and variances μ_j and σ_j are not identifiable but this does not affect the graph G .



Three examples of nonparanormals.

We can estimate G using a two stage procedure:

1. Estimate each $Z_j = f_j(x_j) = \Phi^{-1}(F_j(x_j))$.
2. Apply the glasso to the Z_j 's.

Let $\widehat{f}_j(x_j) = \Phi^{-1}(\widehat{F}_j(x_j))$. The usual empirical $\widehat{F}_j(x_j)$ will not work if d increases with n . We use a Winsorized version:

$$\widetilde{F}_j(x) = \begin{cases} \delta_n & \text{if } \widehat{F}_j(x) < \delta_n \\ \widehat{F}_j(x) & \text{if } \delta_n \leq \widehat{F}_j(x) \leq 1 - \delta_n \\ (1 - \delta_n) & \text{if } \widehat{F}_j(x) > 1 - \delta_n, \end{cases}$$

where

$$\delta_n \equiv \frac{1}{4n^{1/4}\sqrt{\pi \log n}}.$$

This choice of δ_n provides the right bias-variance balance so that we can achieve the desired rate of convergence in our estimate of Ω and the associated undirected graph G . Now compute the sample covariance S_n of the Normalized variables: $Z_j = \widehat{f}_j(X_j) = \Phi^{-1}(\widetilde{F}_j(X_j))$. Finally, apply the glasso to S_n . Let S_n^* be the covariance using the true f_j 's.

Suppose that $d \leq n^\xi$. For large n ,

$$\mathbb{P} \left(\max_{jk} |S_n(j, k) - S_n^*(j, k)| > \epsilon \right) \leq \frac{c_1 d}{(n\epsilon^2)^{2\xi}} + \frac{c_1 d}{(n\epsilon^2)^{c_5 \xi - 1}} + c_3 \exp \left(- \frac{c_4 n^{1/2} \epsilon^2}{\log d \log^2 n} \right)$$

and hence

$$\max_{jk} |S_n(j, k) - S_n^*(j, k)| = O_P \left(\sqrt{\frac{\log d \log^2 n}{n^{1/2}}} \right).$$

Suppose (unrealistically) that $X^{(i)} \sim \text{NPN}(\mu_0, \Sigma_0, f_0)$, and let $\Omega_0 = \Sigma_0^{-1}$. If

$$\lambda_n \asymp \sqrt{\frac{\log d \log^2 n}{n^{1/2}}}$$

then $\|\widehat{\Omega}_n - \Omega_0\|_F = O_P \left(\sqrt{\frac{(s+d) \log d \log^2 n}{n^{1/2}}} \right)$ and

$\|\widehat{\Omega}_n - \Omega_0\|_2 = O_P \left(\sqrt{\frac{s \log d \log^2 n}{n^{1/2}}} \right)$ where s is the sparsity level. Under extra conditions we get sparsistency:

$$\mathbb{P} \left(\text{sign}(\widehat{\Sigma}_n(j, k)) = \text{sign}(\Sigma_0(j, k) \quad \text{for all } j, k) \right) \rightarrow 1.$$

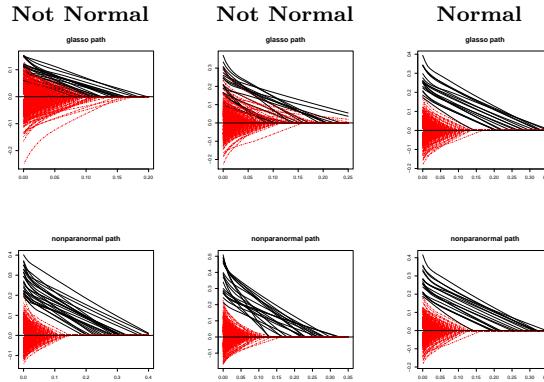
Now suppose (more realistically) P is not NPN. Let $R(\widehat{f}, \widehat{\Sigma})$ denote risk (expected log-likelihood). Let $d \leq e^{n^\xi}$ for $\xi < 1$ and let

$$\mathcal{M}_n = \{f : f_j \text{ monotone, } \|f_j\|_\infty \leq C \sqrt{\log n}\},$$

$$\mathcal{C}_n = \{\Omega : \|\Omega^{-1}\|_1 \leq L_n\}$$

with $L_n = o(n^{(1-\xi)/2}/\sqrt{\log n})$. Then

$$R(\widehat{f}, \widehat{\Omega}) - \inf_{f, \Omega} R(f, \Omega) = o_P(1).$$



$n = 500, p = 40$

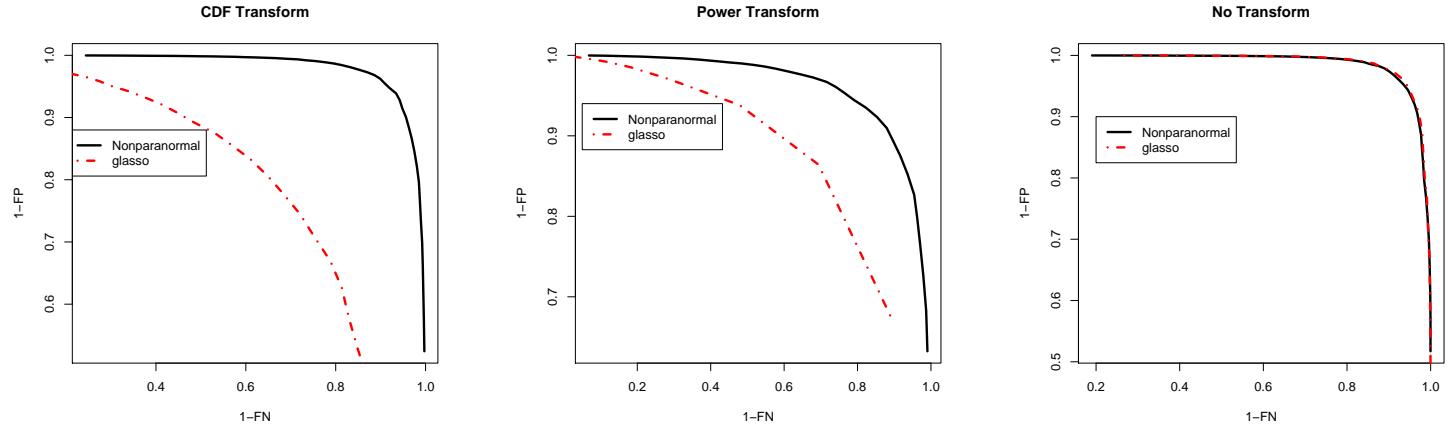


Figure 6: ROC curves for sample sizes $n = 200$.

The *Skeptic* is a more robust version (Spearman/Kendall Estimators Pre-empt Transformations to Infer Correlation). Set $\widehat{S}_{jk} = \sin\left(\frac{\pi}{2}\widehat{\tau}_{jk}\right)$ where

$$\widehat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \leq s < t \leq n} \text{sign}\left((X_s(j) - X_t(j))(X_s(k) - X_t(k))\right).$$

Then (with $d \geq n$)

$$\mathbb{P}\left(\max_{jk} |\widehat{S}_{jk} - \Sigma_{jk}| > 2.45\pi\sqrt{\frac{\log d}{n}}\right) \leq \frac{1}{d}.$$

As in Yuan (2010), let

$$\mathcal{M} = \left\{ \Omega : \Omega \succ 0, \|\Omega\|_1 \leq \kappa, \frac{1}{c} \leq \lambda_{\min}(\Omega) \leq \lambda_{\max}(\Omega) < c, \deg(\Omega) \leq M \right\}.$$

Then, for all $1 \leq q \leq \infty$,

$$\sup_{\Omega \in \mathcal{M}} \|\widehat{\Omega} - \Omega\|_q = O_P\left(M\sqrt{\frac{\log d}{n}}\right).$$

From Yuan, this implies that the Skeptic is minimax rate optimal. Now plug \widehat{S} into glasso. See Liu, Han, Yuan, Lafferty and Wasserman arXiv:1202.2169 for numerical experiments and theoretical results.

Forests. Yet another approach is based on *forests*. A tractable family of graphs are *forests*. A graph F is a forest if it contains no cycles. If F is a d -node undirected forest with vertex set $V_F = \{1, \dots, d\}$ and edge set $E_F \subset \{1, \dots, d\} \times \{1, \dots, d\}$, the number of edges satisfies $|E_F| < d$. Suppose that P is Markov to F and has density p . Then p can be written as

$$p(x) = \prod_{(i,j) \in E_F} \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \prod_{k \in V_F} p(x_k), \quad (11)$$

where each $p(x_i, x_j)$ is a bivariate density, and each $p(x_k)$ is a univariate density. Using (11), we have

$$\begin{aligned} & \mathbb{E} \log p(X) \\ &= - \int p(x) \left(\sum_{(i,j) \in E_F} \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)} + \sum_{k \in V_F} \log(p(x_k)) \right) dx \\ &= - \sum_{(i,j) \in E_F} \int p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)} dx_i dx_j - \sum_{k \in V_F} \int p(x_k) \log p(x_k) dx_k \\ &= - \sum_{(i,j) \in E_F} I(X_i; X_j) + \sum_{k \in V_F} H(X_k), \end{aligned} \quad (13)$$

where

$$I(X_i; X_j) \equiv \int p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)} dx_i dx_j \quad (14)$$

is the mutual information between the pair of variables X_i, X_j and

$$H(X_k) \equiv -ds \int p(x_k) \log p(x_k) dx_k \quad (15)$$

is the entropy.

The optimal forest F^* can be found by minimizing the right hand side of (13). Since the entropy term $H(X) = \sum_k H(X_k)$ is constant across all forests, this can be recast as the problem of finding the maximum weight spanning forest for a weighted graph, where the weight $w(i, j)$ of the edge connecting nodes i and j is $I(X_i; X_j)$. Kruskal's algorithm (Kruskal 1956) is a greedy algorithm that is guaranteed to find a maximum weight spanning tree of a weighted graph. In the setting of density estimation, this procedure was proposed by Chow and Liu (1968) as a way of constructing a tree approximation to a distribution. At each stage the algorithm adds an edge connecting that pair of variables with maximum mutual information among all pairs not yet visited by the algorithm, if doing so does not form a cycle. When stopped early, after $k < d$ edges have been added, it yields the best k -edge weighted forest.

Of course, the above procedure is not practical since the true density $p(x)$ is unknown. In applications, we parameterize bivariate and univariate distributions to be $p_{\theta_{ij}}(x_i, x_j)$ and

Chow-Liu Algorithm for Learning Forest Graphs

Initialize $E^{(0)} = \emptyset$ and the desired forest size $K < d$.

Calculate the mutual information matrix $\widehat{M} = [\widehat{I}_n(X_i, X_j)]$ according to (16).

For $k = 1, \dots, K$

- (a) $(i^{(k)}, j^{(k)}) \leftarrow \operatorname{argmax}_{(i,j)} \widehat{M}(i, j)$ such that $E^{(k-1)} \cup \{(i^{(k)}, j^{(k)})\}$ does not contain a cycle.
- (b) $E^{(k)} \leftarrow E^{(k-1)} \cup \{(i^{(k)}, j^{(k)})\}$.

Output the obtained edge set $E^{(K)}$.

$p_{\theta_k}(x_k)$. We replace the population mutual information $I(X_i; X_j)$ in (13) by the plug-in estimate $\widehat{I}_n(X_i, X_j)$, defined as

$$\widehat{I}_n(X_i, X_j) = \int p_{\widehat{\theta}_{ij}}(x_i, x_j) \log \frac{p_{\widehat{\theta}_{ij}}(x_i, x_j)}{p_{\widehat{\theta}_i}(x_i) p_{\widehat{\theta}_j}(x_j)} dx_i dx_j \quad (16)$$

where $\widehat{\theta}_{ij}$ and $\widehat{\theta}_k$ are maximum likelihood estimates. Given this estimated mutual information matrix $\widehat{M} = [\widehat{I}_n(X_i, X_j)]$, we can apply Kruskal's algorithm (equivalently, the Chow-Liu algorithm) to find the best forest structure \widehat{F} . The detailed algorithm is described in the following:

Example 14 (Learning Gaussian maximum weight spanning tree) For Gaussian data $X \sim N(\mu, \Sigma)$, we know that the mutual information between two variables are

$$I(X_i; X_j) = -\frac{1}{2} \log (1 - \rho_{ij}^2), \quad (17)$$

where ρ_{ij} is the correlation between X_i and X_j . To obtain an empirical estimator, we simply plug-in the sample correlation $\widehat{\rho}_{ij}$. Once the mutual information matrix is calculated, we could apply the Chow-Liu algorithm to get the maximum weight spanning tree.

Example 15 (Graphs for Equities Data) We collect the daily closing prices were obtained for 452 stocks that were consistently in the S&P 500 index between January 1, 2003 through January 1, 2011. This gave us altogether 2,015 data points, each data point corresponds to the vector of closing prices on a trading day. With $S_{t,j}$ denoting the closing price of stock j on day t , we consider the variables $X_{t,j} = \log(S_{t,j}/S_{t-1,j})$ and build graphs over the indices j . We simply treat the instances X_t as independent replicates, even though they form a time series. We truncate every stock so that its data points are within six times the

mean absolute deviation from the sample average. In Figure 7(a) we show boxplots for 10 randomly chosen stocks. It can be seen that the data contains outliers even after truncation; the reasons for these outliers includes splits in a stock, which increases the number of shares. In Figure 7(b) we show the boxplots of the data after the nonparanormal transformation (the details of nonparanormal transformation will be explained in the nonparametric graphical model chapter). In this analysis, we use the subset of the data between January 1, 2003 to January 1, 2008, before the onset of the “financial crisis.” There are altogether $n = 1,257$ data points and $d = 452$ dimensions.

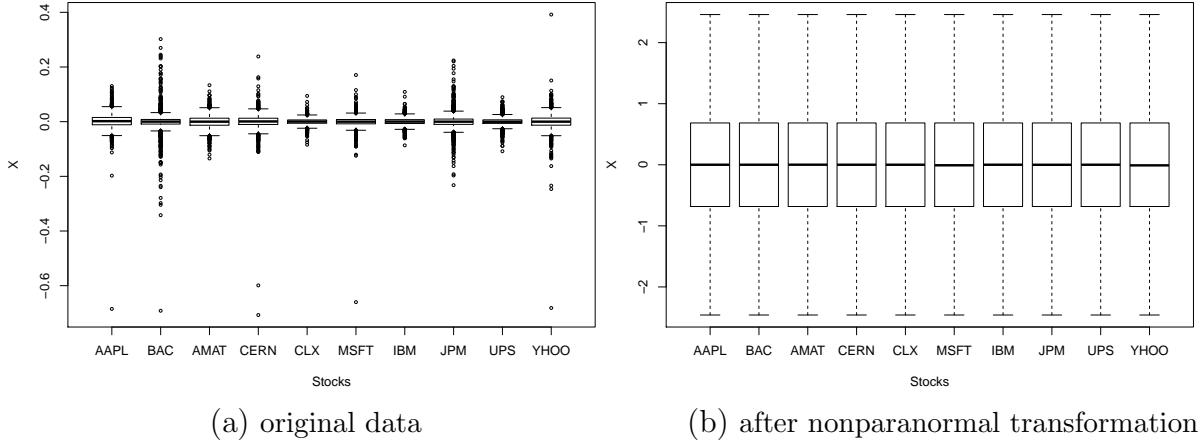


Figure 7: Boxplots of $X_t = \log(S_t/S_{t-1})$ for 10 stocks. As can be seen, the original data has many outliers, which is addressed by the nonparanormal transformation on the re-scaled data (right).

The 452 stocks are categorized into 10 Global Industry Classification Standard (GICS) sectors, including **Consumer Discretionary** (70 stocks), **Energy** (37 stocks), **Financials** (74 stocks), **Consumer Staples** (35 stocks), **Telecommunications Services** (6 stocks), **Health Care** (46 stocks), **Industrials** (59 stocks), **Information Technology** (64 stocks), **Materials** (29 stocks), and **Utilities** (32 stocks). It is expected that stocks from the same GICS sectors should tend to be clustered together, since stocks from the same GICS sector tend to interact more with each other. In the graphs shown below, the nodes are colored according to the GICS sector of the corresponding stock.

With the Gaussian assumption, we directly apply Chow-Liu algorithm to obtain a full spanning tree of $d - 1 = 451$ edges. The resulting graph is shown in Figure 8. We see that the stocks from the same GICS sector are clustered very well.

To get a nonparametric version, we can just an nonparametric estimate of the mutual information. But for that matter, we might as well put in any measure of association such as

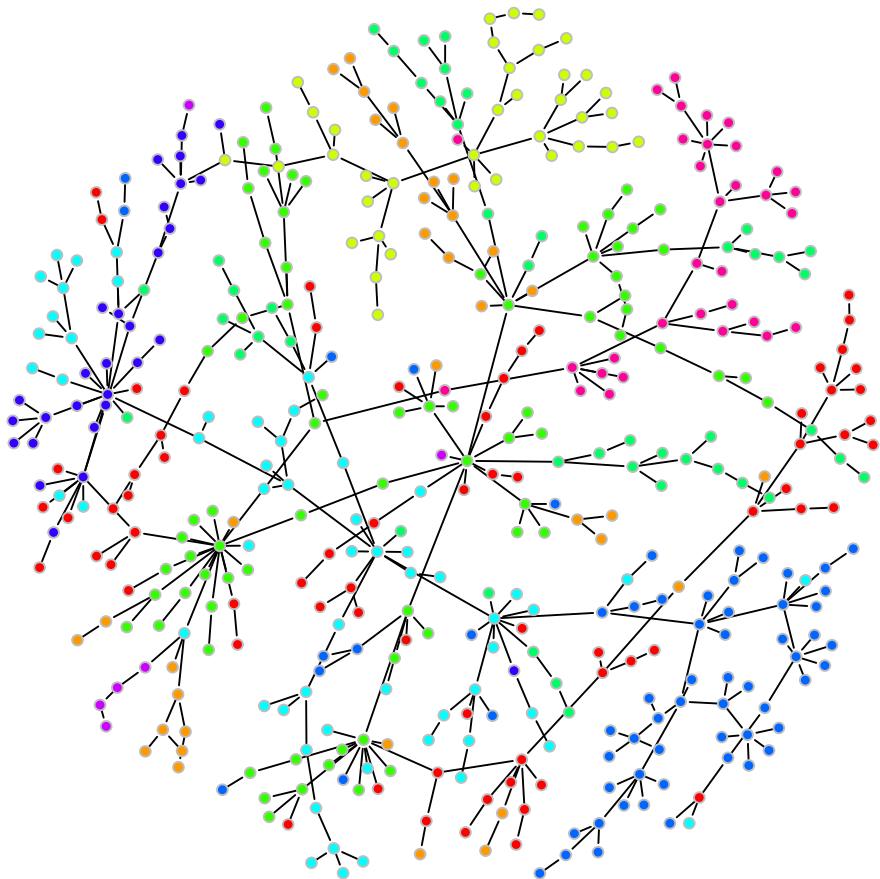


Figure 8: Tree graph learned from S&P 500 stock data from Jan. 1, 2003 to Jan. 1, 2008. The graph is estimated using the Chow-Liu algorithm under the Gaussian model. The nodes are colored according to their GICS sector categories.

distance correlation. In that case, we see that a forest is just a correlation graph without cycles.

4 A Deeper Look At Conditional Independence Graphs

In this section, we will take a closer look at conditional independence graphs.

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E , and let A , B , and C be subsets of vertices. We say that C separates A and B if every path from a node in A to a node in B passes through a node in C . Now consider a random vector $X = (X(1), \dots, X(d))$ where X_j corresponds to node j in the graph. If $A \subset \{1, \dots, d\}$ then we write $X_A = (X(j) : j \in A)$.

4.1 Markov Properties

A probability distribution P for a random vector $X = (X(1), \dots, X(d))$ may satisfy a range of different *Markov properties* with respect to a graph $G = (V, E)$:

Definition 16 (Global Markov Property) *A probability distribution P for a random vector $X = (X(1), \dots, X(d))$ satisfies the global Markov property with respect to a graph G if for any disjoint vertex subsets A , B , and C such that C separates A and B , the random variables X_A are conditionally independent of X_B given X_C .*

The set of distributions that is globally Markov with respect to G is denoted by $\mathcal{P}(G)$.

Definition 17 (Local Markov Property) *A probability distribution P for a random vector $X = (X(1), \dots, X(d))$ satisfies the local Markov property with respect to a graph G if the conditional distribution of a variable given its neighbors is independent of the remaining nodes. That is, let $N(s) = \{t \in V \mid (s, t) \in E\}$ denote the set of neighbors of a node $s \in V$. Then the local Markov property is that*

$$p(x_s \mid x_t, t \neq s) = p(x_s \mid x_t, t \in N(s)) \quad (18)$$

for each node s .

Definition 18 (Pairwise Markov Property) *A probability distribution P for a random vector $X = (X(1), \dots, X(d))$ satisfies the pairwise Markov property with respect to a graph G if for any pair of non-adjacent nodes $s, t \in V$, we have*

$$X_s \perp\!\!\!\perp X_t \mid X_{V \setminus \{s, t\}}. \quad (19)$$

Consider for example the graph in Figure 9. Here the set C separates A and B . Thus, a distribution that satisfies the global Markov property for this graph must have the property that the random variables in A are conditionally independent of the random variables in B given the random variables in C . This is seen to generalize the usual Markov property for simple chains, where $X_A \rightarrow X_C \rightarrow X_B$ forms a Markov chain in case X_A and X_B are independent given X_C . A distribution that satisfies the global Markov property is said to be a *Markov random field* or *Markov network* with respect to the graph. The *local Markov property* is depicted in Figure 10.

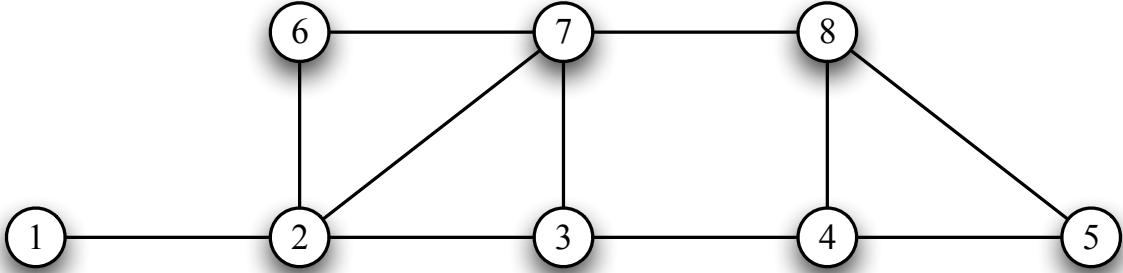


Figure 9: An undirected graph. $C = \{3, 7\}$ separates $A = \{1, 2\}$ and $B = \{4, 8\}$.

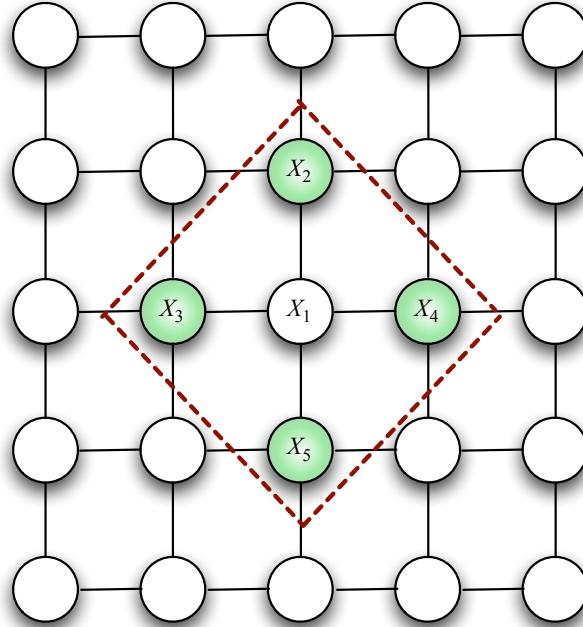


Figure 10: The local Markov property: Conditioned on its four neighbors X_2 , X_3 , X_4 , and X_5 , node X_1 is independent of the remaining nodes in the graph.

From the definitions, the relationships of different Markov properties can be characterized as:

Proposition 19 *For any undirected graph G and any distribution P , we have*

global Markov property \implies local Markov property \implies pairwise Markov property.

Proof. The global Markov property implies the local Markov property because for each node $s \in V$, its neighborhood $N(s)$ separates $\{s\}$ and $V \setminus \{N(s) \cup \{s\}\}$. Assume next that the local Markov property holds. Any t that is not adjacent to s is an element of $t \in V \setminus \{N(s) \cup \{s\}\}$. Therefore

$$N(s) \cup [(V \setminus \{N(s) \cup \{s\}\}) \setminus \{t\}] = V \setminus \{s, t\}, \quad (20)$$

and it follows from the local Markov property that

$$X_s \perp\!\!\!\perp X_{V \setminus \{N(s) \cup \{s\}\}} | X_{V \setminus \{s, t\}}. \quad (21)$$

This implies $X_s \perp\!\!\!\perp X_t | X_{V \setminus \{s, t\}}$, which is the pairwise Markov property. \square

The next theorem, due to Pearl (1986), provides a sufficient condition for equivalence.

Theorem 20 *Suppose that, for all disjoint subsets $A, B, C, D \subset V$,*

$$\text{if } X_A \perp\!\!\!\perp X_B | X_{C \cup D} \text{ and } X_A \perp\!\!\!\perp X_C | X_{B \cup D}, \text{ then } X_A \perp\!\!\!\perp X_{B \cup C} | X_D, \quad (22)$$

then the global, local, and pairwise Markov properties are equivalent.

Proof. It is enough to show that the pairwise Markov property implies the global Markov property under the given condition. Let $S, A, B \subset V$ with S separating A from B in the graph G . Without loss of generality both A and B are assumed to be non-empty. The proof can be carried out using backward induction on the number of nodes in S , denoted by $m = |S|$. Let $d = |V|$, for the base case, if $m = d - 1$ then both A and B only consist of single vertex and the result follows from pairwise Markov property.

Now assume that $m < d - 1$ and separation implies conditional independence for all separating sets S with more than m nodes. We proceed in two cases: (i) $A \cup B \cup S = V$ and (ii) $A \cup B \cup S \subset V$.

For case (i), we know that at least one of A and B must have more than one element. Without loss of generality, we assume A has more than one element. If $s \in A$, then $S \cup \{s\}$ separates $A \setminus \{s\}$ from B and also $S \cup (A \setminus \{s\})$ separates s from B . Thus by the induction hypothesis

$$X_{A \setminus \{s\}} \perp\!\!\!\perp X_B | X_{S \cup \{s\}} \text{ and } X_s \perp\!\!\!\perp X_B | S \cup (A \setminus \{s\}). \quad (23)$$

Now the condition (22) implies $X_A \perp\!\!\!\perp X_B | X_S$. For case (ii), we could choose $s \in V \setminus (A \cup B \cup S)$. Then $S \cup \{s\}$ separates A and B , implying $A \perp\!\!\!\perp B | S \cup \{s\}$. We then proceed in two cases, either $A \cup S$ separates B from s or $B \cup S$ separates A from s . For both cases, the condition (22) implies that $A \perp\!\!\!\perp B | S$. \square

The next proposition provides a stronger condition that implies (22).

Proposition 21 Let $X = (X_1, \dots, X_d)$ be a random vector with distribution P and joint density $p(x)$. If the joint density $p(x)$ is positive and continuous with respect to a product measure, then condition (22) holds.

Proof. Without loss of generality, it suffices to assume that $d = 3$. We want to show that

$$\text{if } X_1 \perp\!\!\!\perp X_2 | X_3 \text{ and } X_1 \perp\!\!\!\perp X_3 | X_2 \text{ then } X_1 \perp\!\!\!\perp \{X_2, X_3\}. \quad (24)$$

Since the density is positive and $X_1 \perp\!\!\!\perp X_2 | X_3$ and $X_1 \perp\!\!\!\perp X_3 | X_2$, we know that there must exist some positive functions $f_{13}, f_{23}, g_{12}, g_{23}$ such that the joint density takes the following factorization:

$$p(x_1, x_2, x_3) = f_{13}(x_1, x_3)f_{23}(x_2, x_3) = g_{12}(x_1, x_2)g_{23}(x_2, x_3). \quad (25)$$

Since the density is continuous and positive, we have

$$g_{12}(x_1, x_2) = \frac{f_{13}(x_1, x_3)f_{23}(x_2, x_3)}{g_{23}(x_2, x_3)}. \quad (26)$$

For each fixed $X_3 = x'_3$, we see that $g_{12}(x_1, x_2) = h(x_1)\ell(x_2)$ where $h(x_1) = f_{13}(x_1, x'_3)$ and $\ell(x_2) = f_{23}(x_2, x'_3)/g_{23}(x_2, x'_3)$. This implies that

$$p(x_1, x_2, x_3) = h(x_1)\ell(x_2)g_{23}(x_2, x_3) \quad (27)$$

and hence $X_1 \perp\!\!\!\perp \{X_2, X_3\}$ as desired. \square

From Proposition 21, we see that for distributions with positive continuous densities, the global, local, and pairwise Markov properties are all equivalent. If a distribution P satisfies global Markov property with respect to a graph G , we say that P is *Markov to G*.

4.2 Clique Decomposition

Unlike a directed graph which encodes a factorization of the joint probability distribution in terms of conditional probability distributions. An undirected graph encodes a factorization of the joint probability distribution in terms of clique potentials. Recall that a *clique* in a graph is a fully connected subset of vertices. Thus, every pair of nodes in a clique is connected by an edge. A clique is a *maximal clique* if it is not contained in any larger clique. Consider, for example, the graph shown in the right plot of Figure 11. The pairs $\{X_4, X_5\}$ and $\{X_1, X_3\}$ form cliques; $\{X_4, X_5\}$ is a maximal clique, while $\{X_1, X_3\}$ is not maximal since it is contained in a larger clique $\{X_1, X_2, X_3\}$.

A set of clique potentials $\{\psi_C(x_C) \geq 0\}_{C \in \mathcal{C}}$ determines a probability distribution that factors with respect to the graph by normalizing:

$$p(x_1, \dots, x_{|V|}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C). \quad (28)$$

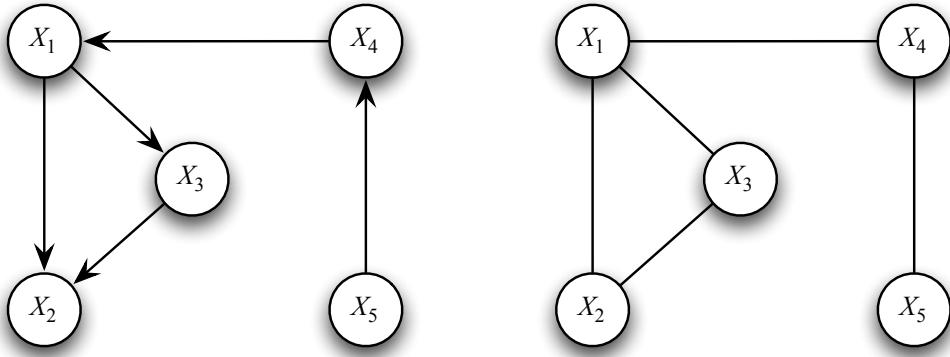


Figure 11: A directed graph encodes a factorization of the joint probability distribution in terms of conditional probability distributions. An undirected graph encodes a factorization of the joint probability distribution in terms of clique potentials.

The *normalizing constant* or *partition function* Z sums (or integrates) over all settings of the random variables:

$$Z = \int_{x_1, \dots, x_{|V|}} \prod_{C \in \mathcal{C}} \psi_C(x_C) dx_1 \dots dx_{|V|}. \quad (29)$$

Thus, the family of distributions represented by the undirected graph in Figure 11 can be written as

$$p(x_1, x_2, x_3, x_4, x_5) = \psi_{1,2,3}(x_1, x_2, x_3) \psi_{1,4}(x_1, x_4) \psi_{4,5}(x_4, x_5). \quad (30)$$

In contrast, the family of distributions represented by the directed graph in Figure 11 can be factored into conditional distributions according to

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_5) p(x_4 | x_5) p(x_1 | x_4) p(x_3 | x_1) p(x_2 | x_1, x_3). \quad (31)$$

Theorem 22 *For any undirected graph $G = (V, E)$, a distribution P that factors with respect to the graph also satisfies the global Markov property on the graph.*

Proof. Let $A, B, S \subset V$ such that S separates A and B . We want to show $X_A \perp\!\!\!\perp X_B | X_S$. For a subset $D \subset V$, we denote G_D to be the subgraph induced by the vertex set D . We define \tilde{A} to be the connectivity components in $G_{V \setminus S}$ which contain A and $\tilde{B} = V \setminus (\tilde{A} \cup S)$. Since A and B are separated by S , they must belong to different connectivity components of $G_{V \setminus S}$ and any clique of G must be a subset of either $\tilde{A} \cup S$ or $\tilde{B} \cup S$. Let \mathcal{C}_A be the set of cliques contained in $\tilde{A} \cup S$, the joint density $p(x)$ takes the following factorization

$$p(x) = \prod_{C \in \mathcal{C}} \psi_C(x_C) = \prod_{C \in \mathcal{C}_A} \psi_C(x_C) \prod_{C \in \mathcal{C} \setminus \mathcal{C}_A} \psi_C(x_C). \quad (32)$$

This implies that $\tilde{A} \perp\!\!\!\perp \tilde{B} | S$ and thus $A \perp\!\!\!\perp B | S$. \square

It is worth remembering that while we think of the set of *maximal cliques* as given in a list, the problem of enumerating the set of maximal cliques in a graph is NP-hard, and the problem of determining the largest maximal clique is NP-complete. However, many graphs of interest in statistical analysis are sparse, with the number of cliques of size $O(|V|)$.

Theorem 22 shows that factoring with respect to a graph implies global Markov property. The next question is, under what conditions the Markov properties imply factoring with respect to a graph. In fact, in the case where P has a positive and continuous density we can show that the pairwise Markov property implies factoring with respect to a graph. Thus all Markov properties are equivalent. The results have been discovered by many authors but is usually referred to as Hammersley and Clifford due to one of their unpublished manuscript in 1971. They proved the result in the discrete case. The following result is usually referred to as the *Hammersley-Clifford theorem*; a proof appears in Besag (1974). The extension to the continuous case is left as an exercise.

Theorem 23 (Hammersley-Clifford-Besag) *Suppose that $G = (V, E)$ is a graph and $X_i, i \in V$ are random variables that take on a finite number of values. If $p(x) > 0$ is strictly positive and satisfies the local Markov property with respect to G , then it factors with respect to G .*

Proof. Let $d = |V|$. By re-indexing the values of X_i , we may assume without loss of generality that each X_i takes on the value 0 with positive probability, and $\mathbb{P}(0, 0, \dots, 0) > 0$. Let $X_{0 \setminus i}$ denote the vector $X_{0 \setminus i} = (X_1, X_2, \dots, X_{i-1}, 0, X_{i+1}, \dots, X_d)$ obtained by setting $X_i = 0$, and let $X_{\setminus i} = (X_1, X_2, \dots, X_{i-1}, X_{i+1}, \dots, X_d)$ denote the vector of all components except X_i . Then

$$\frac{\mathbb{P}(x)}{\mathbb{P}(x_{i \setminus 0})} = \frac{\mathbb{P}(x_i | x_{\setminus i})}{\mathbb{P}(0 | x_{\setminus i})}. \quad (33)$$

Now, let

$$Q(x) = \log \left(\frac{\mathbb{P}(x)}{\mathbb{P}(0)} \right). \quad (34)$$

Then for any $i \in \{1, 2, \dots, d\}$ we have that

$$Q(x) = \log \left(\frac{\mathbb{P}(x)}{\mathbb{P}(0)} \right) \quad (35)$$

$$= \log \left(\frac{\mathbb{P}(0, \dots, x_i, 0, \dots, 0)}{\mathbb{P}(0)} \right) + \log \left(\frac{\mathbb{P}(x)}{\mathbb{P}(0, \dots, x_i, 0, \dots, 0)} \right) \quad (36)$$

$$= \frac{1}{d} \sum_{i=1}^d \left\{ \log \left(\frac{\mathbb{P}(0, \dots, x_i, 0, \dots, 0)}{\mathbb{P}(0)} \right) + \log \left(\frac{\mathbb{P}(x)}{\mathbb{P}(0, \dots, x_i, 0, \dots, 0)} \right) \right\}. \quad (37)$$

Recursively, we obtain

$$Q(x) = \sum_i \phi_i(x_i) + \sum_{i < j} \phi_{ij}(x_i, x_j) + \sum_{i < j < k} \phi_{ijk}(x_i, x_j, x_k) + \cdots + \phi_{12\dots d}(x)$$

for functions ϕ_A that satisfy $\phi_A(x_A) = 0$ if $i \in A$ and $x_i = 0$. Consider node $i = 1$, we have

$$\begin{aligned} Q(x) - Q(x_{0 \setminus i}) &= \log \left(\frac{\mathbb{P}(x_i | x_{\setminus i})}{\mathbb{P}(0 | x_{\setminus i})} \right) \\ &= \phi_1(x_1) + \sum_{i > 1} \phi_{1i}(x_1, x_i) + \sum_{j > i > 1} \phi_{1ij}(x_1, x_i, x_j) + \cdots + \phi_{12\dots d}(x) \end{aligned} \quad (38)$$

depends only on x_1 and the neighbors of node 1 in the graph. Thus, from the local Markov property, if k is not a neighbor of node 1, then the above expression does not depend of x_k . In particular, $\phi_{1k}(x_1, x_k) = 0$, and more generally all $\phi_A(x_A)$ with $1 \in A$ and $k \in A$ are identically zero. Similarly, if i, j are not neighbors in the graph, then $\phi_A(x_A) = 0$ for any A containing i and j . Thus, $\phi_A \neq 0$ only holds for the subsets A that form cliques in the graph. Since it is obvious that $\exp(\phi_A(x)) > 0$, we finish the proof. \square

Since factoring with respect to the graph implies the global Markov property, we may summarize this result as follows:

For positive distributions: global Markov \Leftrightarrow local Markov \Leftrightarrow factored

For strictly positive distributions, the global Markov property, the local Markov property, and factoring with respect to the graph are equivalent.

Thus we can write:

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C) = \frac{1}{Z} \exp \left(\sum_{C \in \mathcal{C}} \log \psi_C(x_C) \right)$$

where \mathcal{C} is the set of all (maximal) cliques in G and Z is the normalization constant. This is called the *Gibbs representation*.

4.3 Directed vs. Undirected Graphs

Directed graphical models are naturally viewed as generative; the graph specifies a straightforward (in principle) procedure for sampling from the underlying distribution. For instance, a sample from a distribution represented from the DAG in left plot of Figure 12 can be

sampled as follows:

$$X_1 \sim P(X_1) \quad (39)$$

$$X_2 \sim P(X_2) \quad (40)$$

$$X_3 \sim P(X_3) \quad (41)$$

$$X_5 \sim P(X_5) \quad (42)$$

$$X_4 | X_1, X_2 \sim P(X_4 | X_1, X_2) \quad (43)$$

$$X_6 | X_3, X_4, X_5 \sim P(X_6 | X_3, X_4, X_5). \quad (44)$$

As long as each of the conditional probability distributions can be efficiently sampled, the full model can be efficiently sampled as well. In contrast, there is no straightforward way to sample from a distribution from the family specified by an undirected graph. Instead one needs something like MCMC.

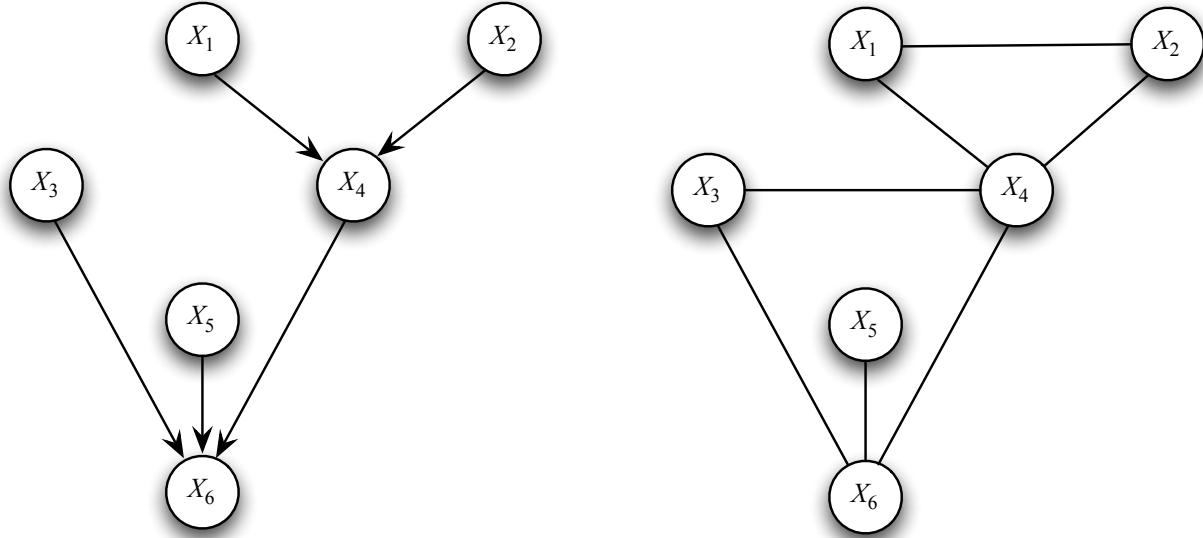


Figure 12: A DAG and its corresponding moral graph. A probability distribution that factors according to a DAG obeys the global Markov property on the undirected moral graph.

Generally, edges must be added to the skeleton of a DAG in order for the distribution to satisfy the global Markov property on the graph. Consider the example in Figure 12. Here the directed model has a distribution

$$p(x_1) p(x_2) p(x_3) p(x_5) p(x_4 | x_1, x_2) p(x_6 | x_3, x_4, x_5). \quad (45)$$

The corresponding undirected graphical model has two maximal cliques, and factors as

$$\psi_{1,2,4}(x_1, x_2, x_4) \psi_{3,4,5,6}(x_3, x_4, x_5, x_6). \quad (46)$$

More generally, let P be a probability distribution that is Markov to a DAG G . We define the *moralized graph* of G as the following:

Definition 24 (Moral graph) *The moral graph M of a DAG G is an undirected graph that contains an undirected edge between two nodes X_i and X_j if (i) there is a directed edge between X_i and X_j in G , or (ii) X_i and X_j are both parents of the same node.*

Theorem 25 *If a probability distribution factors with respect to a DAG G , then it obeys the global Markov property with respect to the undirected moral graph of G .*

Proof. Directly follows from the definition of Bayesian networks and Theorem 22. \square

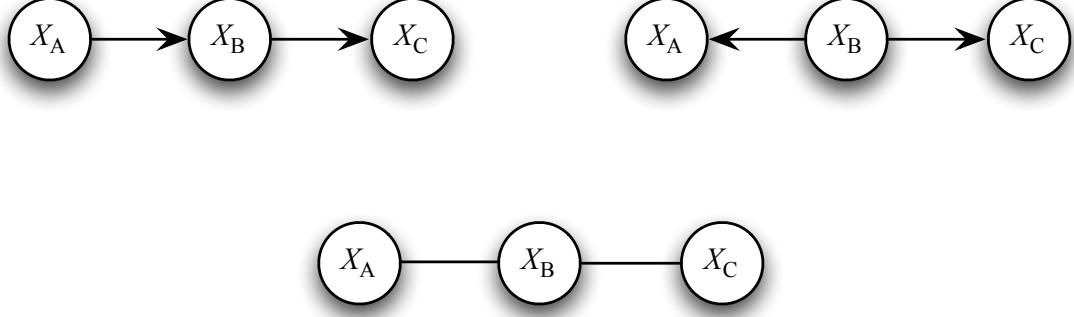


Figure 13: These three graphs encode distributions with identical independence relations. Conditioned on variable X_C , the variables X_A and X_B are independent; thus C separates A and B in the undirected graph.

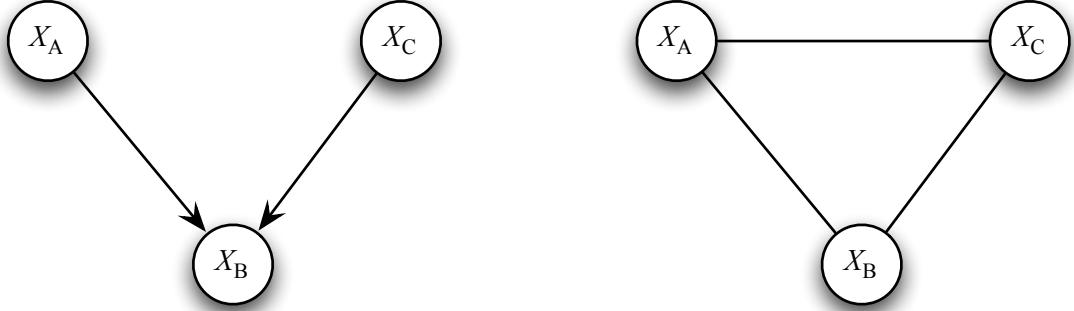


Figure 14: A directed graph whose conditional independence properties can not be perfectly expressed by its undirected moral graph. In the directed graph, the node C is a collider; therefore, X_A and X_B are not independent conditioned on X_C . In the corresponding moral graph, A and B are not separated by C . However, in the directed graph, we have the independence relationship $X_A \perp\!\!\!\perp X_B$, which is missing in the moral graph.

Example 26 (Basic Directed and Undirected Graphs) *To illustrate some basic cases, consider the graphs in Figure 13. Each of the top three graphs encodes the same family of probability*

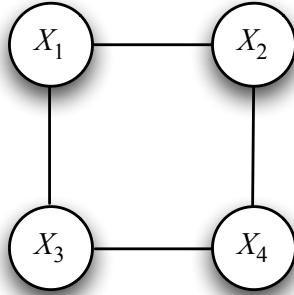


Figure 15: This undirected graph encodes a family of distributions that cannot be represented by a directed graph on the same set of nodes.

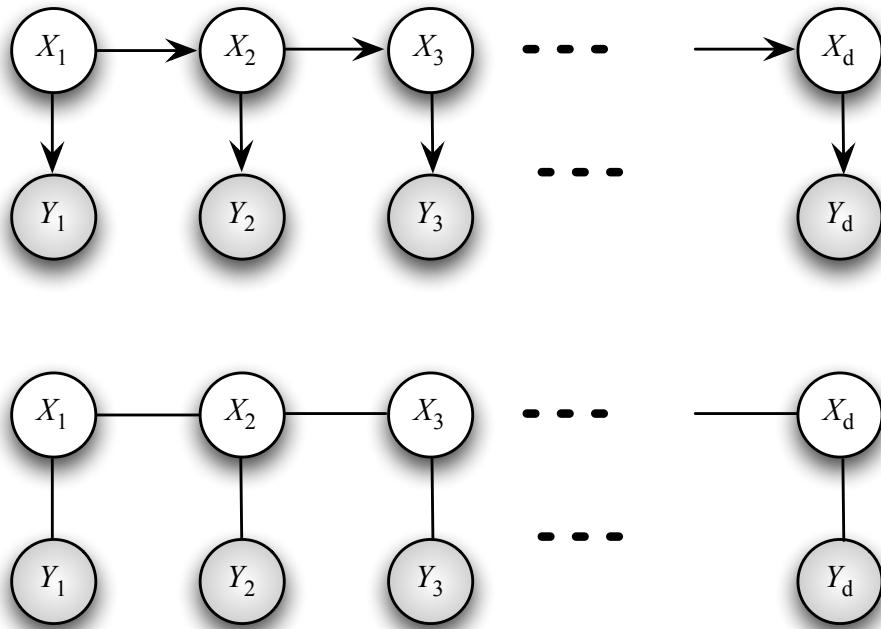


Figure 16: The top graph is a directed graph representing a *hidden Markov model*. The shaded nodes are observed, but the unshaded nodes, representing states in a latent Markov chain, are unobserved. Replacing the directed edges by undirected edges (bottom) does not change the independence relations.

distributions. In the two directed graphs, by d-separation the variables X_A and X_B are independent conditioned on the variable X_C . In the corresponding undirected graph, which simply removes the arrows, node C separates A and B .

The two graphs in Figure 14 provide an example of a directed graph which encodes a set of conditional independence relationships that can not be perfectly represented by the corresponding moral graph. In this case, for the directed graph the node C is a collider, and deriving an equivalent undirected graph requires joining the parents by an edge. In the corresponding undirected graph, A and B are not separated by C . However, in the directed graph, X_A and X_B are marginally independent, such an independence relationship is lost in the moral graph. Conversely, Figure 15 provides an undirected graph over four variables. There is no directed graph over four variables that implies the same set of conditional independence properties.

The upper plot in Figure 16 shows the directed graph underlying a hidden Markov model. There are no colliders in this graph, and therefore the undirected skeleton represents an equivalent set of independence relations. Thus, hidden Markov models are equivalent to hidden Markov fields with an underlying tree graph.

4.4 Faithfulness

The set of all distributions that are Markov to a graph G is denoted by $\mathcal{P}(G)$. To understand $\mathcal{P}(G)$ more clearly, we introduce some more notation. Given a distribution P let $\mathcal{I}(P)$ denote all conditional independence statements that are true for P . For example, if P has density p and $p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2)$ then $\mathcal{I}(P) = \{X_1 \perp\!\!\!\perp X_2\}$. On the other hand, if $p(x_1, x_2, x_3) = p(x_1)p(x_2, x_3)$ then

$$\mathcal{I}(P) = \left\{ X_1 \perp\!\!\!\perp X_2, X_1 \perp\!\!\!\perp X_3, X_1 \perp\!\!\!\perp X_2|X_3, X_1 \perp\!\!\!\perp X_3|X_2 \right\}.$$

Similarly, given a graph G let $\mathcal{I}(G)$ denote all independence statements implied by the graph. For example, if G is the graph in Figure 15, then

$$\mathcal{I}(G) = \left\{ X_1 \perp\!\!\!\perp X_4 | \{X_2, X_3\}, X_2 \perp\!\!\!\perp X_3 | \{X_1, X_4\} \right\}.$$

From definition, we could write $\mathcal{P}(G)$ as

$$\mathcal{P}(G) = \left\{ P : \mathcal{I}(G) \subseteq \mathcal{I}(P) \right\}. \quad (47)$$

This result often leads to confusion since you might have expected that $\mathcal{P}(G)$ would be equal to $\{P : \mathcal{I}(G) = \mathcal{I}(P)\}$. But this is incorrect. For example, consider the undirected graph $X_1 — X_2$, in this case, $\mathcal{I}(G) = \emptyset$ and $\mathcal{P}(G)$ consists of all distributions $p(x_1, x_2)$. Since, $\mathcal{P}(G)$ consists of all distributions, it also includes distributions of the form $p_0(x_1, x_2) = p_0(x_1)p_0(x_2)$. For such a distribution we have $\mathcal{I}(P_0) = \{X_1 \perp\!\!\!\perp X_2\}$. Hence, $\mathcal{I}(G)$ is a strict subset of $\mathcal{I}(P_0)$.

In fact, you can think of $\mathcal{I}(G)$ as the set of independence statements that are common to all $P \in \mathcal{P}(G)$. In other words,

$$\mathcal{I}(G) = \bigcap \left\{ \mathcal{I}(P) : P \in \mathcal{P}(G) \right\}. \quad (48)$$

Every $P \in \mathcal{P}(G)$ has the independence properties in $\mathcal{I}(G)$. But some P 's in $\mathcal{P}(G)$ might have extra independence properties.

We say that P is *faithful* to G if $\mathcal{I}(P) = \mathcal{I}(G)$. We define

$$\mathcal{F}(G) = \left\{ P : \mathcal{I}(G) = \mathcal{I}(P) \right\} \quad (49)$$

and we note that $\mathcal{F}(G) \subset \mathcal{P}(G)$. A distribution P that is in $\mathcal{P}(G)$ but is not in $\mathcal{F}(G)$ is said to be *unfaithful with respect to G* . The independence relation expressed by G are correct for such a P . It's just that P has extra independence relations not expressed by G . A distribution P is also Markov to some graph. For example, any distribution is Markov to the complete graph. But there exist distributions P that are not faithful to any graph. This means that there will be some independence relations of P that cannot be expressed using a graph.

Example 27 *The directed graph in Figure 14 implies that $X_A \perp\!\!\!\perp X_B$ but that X_A and X_B are not independent given X_C . There is no undirected graph G for (X_A, X_B, X_C) such that $\mathcal{I}(G)$ contains $X_A \perp\!\!\!\perp X_B$ but excludes $X_A \perp\!\!\!\perp X_B | X_C$. The only way to represent P is to use the complete graph. Then P is Markov to G since $\mathcal{I}(G) = \emptyset \subset \mathcal{I}(P) = \{X_A \perp\!\!\!\perp X_B\}$ but P is unfaithful to G since it has an independence relation not represented by G , namely, $\{X_A \perp\!\!\!\perp X_B\}$.*

Example 28 (Unfaithful Gaussian distribution) . Let $\xi_1, \xi_2, \xi_3 \sim N(0, 1)$ be independent.

$$X_1 = \xi_1 \quad (50)$$

$$X_2 = aX_1 + \xi_2 \quad (51)$$

$$X_3 = bX_2 + cX_1 + \xi_3 \quad (52)$$

where a, b, c are nonzero. See Figure 17. Now suppose that $c = -\frac{b(a^2+1)}{a}$. Then

$$\text{Cov}(X_2, X_3) = \mathbb{E}(X_2 X_3) - \mathbb{E}X_2 \mathbb{E}X_3 \quad (53)$$

$$= \mathbb{E}(X_2 X_3) \quad (54)$$

$$= \mathbb{E}[(aX_1 + \xi_2)(bX_2 + cX_1 + \xi_3)] \quad (55)$$

$$= \mathbb{E}[(a\xi_1 + \xi_2)(b(a\xi_1 + \xi_2) + cX_1 + \xi_3)] \quad (56)$$

$$= (a^2b + ac)\mathbb{E}\xi_1^2 + b\mathbb{E}\xi_2^2. \quad (57)$$

$$= a^2b + ac + b = 0. \quad (58)$$

Thus, we know that $X_2 \perp\!\!\!\perp X_3$. We would like to drop the edge between X_2 and X_3 . But this would imply that $X_2 \perp\!\!\!\perp X_3 | X_1$ which is not true.

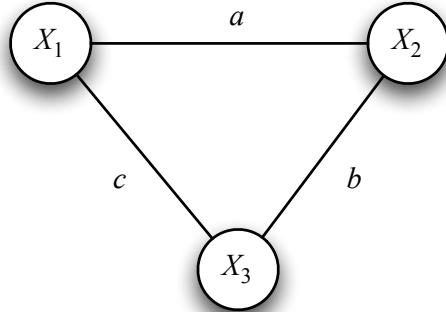


Figure 17: An unfaithful Gaussian distribution.

Generally, the set of unfaithful distributions $\mathcal{P}(G) \setminus \mathcal{F}(G)$ is a small set. In fact, it has Lebesgue measure zero if we restrict ourselves to nice parametric families. However, these unfaithful distributions are scattered throughout $\mathcal{P}(G)$ in a complex way; see Figure 18.

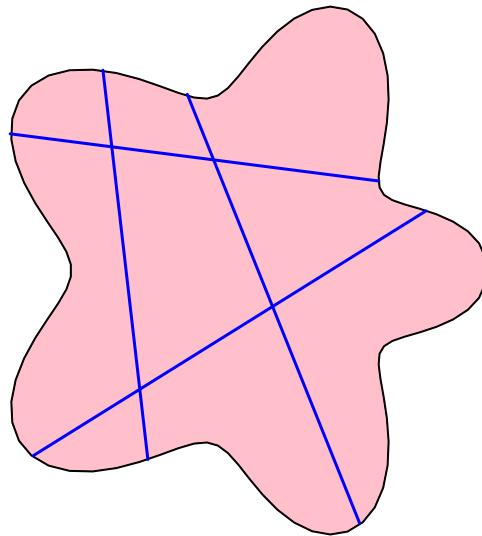


Figure 18: The blob represents the set $\mathcal{P}(G)$ of distributions that are Markov to some graph G . The lines are a stylized representation of the members of $\mathcal{P}(G)$ that are not faithful to G . Hence the lines represent the set $\mathcal{P}(G) \setminus \mathcal{F}(G)$. These distributions have extra independence relations not captured by the graph G . The set $\mathcal{P}(G) \setminus \mathcal{F}(G)$ is small but these distributions are scattered throughout $\mathcal{P}(G)$.

References

Gretton, Song, Fukumizu, Scholkopf, Smola (2008). A Kernel Statistical Test of Independence. NIPS.

Lyons, Russell. (2013). Distance covariance in metric spaces. *The Annals of Probability*, 41, 3284-3305.

Szekely, Gabor J., Maria L. Rizzo, and Nail K. Bakirov. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35, 2769-2794.

Chapter 18

Directed Graphical Models

Graphs give a powerful way of representing independence relations and computing conditional probabilities among a set of random variables. In a directed graphical model, the probability of a set of random variables factors into a product of conditional probabilities, one for each node in the graph.

18.1 Introduction

A graphical model is a probabilistic model for which the conditional independence structure is encoded in a graph. In a graphical model, vertices (or nodes) represent random variables, and the edges encode conditional independence relations among the associated vertices. The graph characterizes the way in which the joint distribution factors into the product of many small components, each of which contains only a subset of variables. In this chapter, we introduce *directed graphical models*, in which the edges of the graph have directions (or arrows). An example of a directed graphical model is shown in Figure 18.1. In the next chapter we introduce *undirected graphical models*, in which the edges carry no directional information.

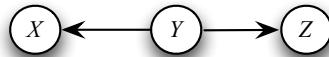


Figure 18.1. A directed graph with vertices $V = \{X, Y, Z\}$ and edges $E = \{(Y, X), (Y, Z)\}$.

Before getting into details, let us recall the definition of conditional independence.

Let X , Y and Z be random variables. X and Y are *conditionally independent* given Z , written $X \perp\!\!\!\perp Y | Z$, if

$$p(x, y|z) = p(x|z)p(y|z). \quad (18.1)$$

for all x , y and z .

Intuitively, this means that, once Z is known, Y provides no extra information about X . An equivalent definition is that $p(x|y, z) = p(x|z)$ for all x , y and z .

Directed graphs are useful for representing conditional independence relations among variables. They can also be used to represent causal relationships. Some people use the phrase *Bayesian network* to refer to a directed graph endowed with a probability distribution. This is a poor choice of terminology. Statistical inference for directed graphs can be performed using either frequentist or Bayesian methods, so it is misleading to call them Bayesian networks.

18.2 Directed Acyclic Graphs (DAGs)

A *directed graph* G consists of a set of vertices V and an edge set E of ordered pairs of vertices. For our purposes, each vertex corresponds to a random variable. If $(Y, X) \in E$ then there is an arrow pointing from Y to X . See Figure 18.1. One thing to note is that a node here can either represent a scalar random variable or a random vector.

If an arrow connects two variables X and Y (in either direction) we say that X and Y are *adjacent*. If there is an arrow from X to Y then X is a *parent* of Y and Y is a *child* of X . The set of all parents of X is denoted by π_X or $\pi(X)$. A *directed path* between two variables is a set of arrows all pointing in the same direction linking one variable to the other such as the chain shown in Figure 18.2:



Figure 18.2. A chain graph with a directed path.

A sequence of adjacent vertices starting with X and ending with Y but ignoring the direction of the arrows is called an *undirected path*. The sequence $\{X, Y, Z\}$ in Figure 18.1 is an undirected path. X is an *ancestor* of Y if there is a directed path from X to Y (or $X = Y$). We also say that Y is a *descendant* of X .

There are three basic connection configurations of three-node subgraphs and larger graphs can be constructed using these three basic configurations. A configuration of the form as in Figure 18.3(a) is called a *collider* at Y (head-to-head connection). A configuration not of that form is called a *non-collider* (head-to-tail and tail-to-tail connections), for example, Figure 18.3(b) and Figure 18.3(c).

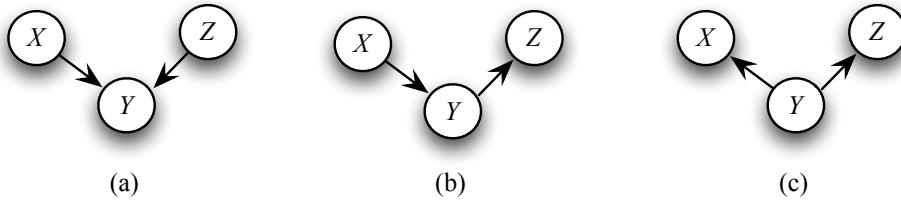


Figure 18.3. (a) a collider at Y ; (b), (c) non-colliders.

The collider property is path dependent. In Figure 18.4, Y is a collider on the path $\{X, Y, Z\}$ but it is a non-collider on the path $\{X, Y, W\}$.

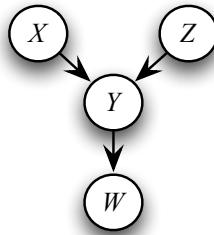


Figure 18.4. A collider with a descendant.

When the variables pointing into a collider are not adjacent, we say that the collider is *unshielded*. A directed path that starts and ends at the same variable is called a *cycle*. A directed graph is *acyclic* if it has no cycles. In this case we say that the graph is a *directed acyclic graph* or *DAG*. From now on, we only deal with directed acyclic graphs since it is very difficult to provide a coherent probability semantics over graphs with directed cycles.

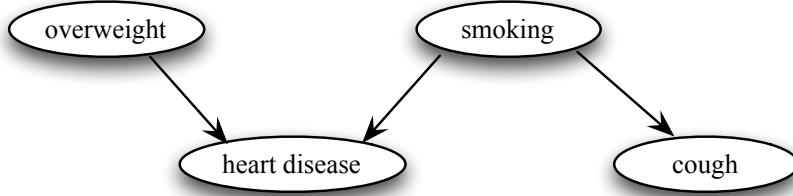
18.3 Probability and DAGs

Let G be a DAG with vertices $V = (X_1, \dots, X_d)$. For notational simplicity, we sometimes represent $V = \{1, \dots, d\}$. If P is a distribution for V with probability function $p(x)$, we say that P is *Markov to* G , or that G *represents* P , if

$$p(x) = \prod_{j=1}^d p(x_j \mid \pi_{x_j}) \quad (18.2)$$

where π_{x_j} is the set of parent nodes of X_j . The set of distributions represented by G is denoted by $\mathcal{M}(G)$.

18.3 Example. Figure 18.5 shows a DAG with four variables. The probability function

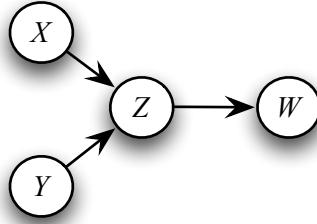
**Figure 18.5.** DAG for Example 18.3.

takes the following decomposition:

$$\begin{aligned} p(\text{overweight}, \text{smoking}, \text{heart disease}, \text{cough}) &= p(\text{overweight}) \times f(\text{smoking}) \\ &\quad \times p(\text{heart disease} \mid \text{overweight}, \text{smoking}) \times p(\text{cough} \mid \text{smoking}). \end{aligned}$$

□

18.4 Example. For the DAG in Figure 18.6, $P \in \mathcal{M}(G)$ if and only if its probability function $p(x)$ has the form $p(x, y, z, w) = p(x)p(y)p(z \mid x, y)p(w \mid z)$. □

**Figure 18.6.** Another DAG.

The following theorem says that $P \in \mathcal{M}(G)$ if and only if the *Markov condition* holds. Roughly speaking, the Markov condition means that every variable W is independent of the “past” given its parents.

18.5 Theorem. For a graph $G = (V, E)$, a distribution $P \in \mathcal{M}(G)$ if and only if the following Markov condition holds: for every variable W ,

$$W \perp\!\!\!\perp \widetilde{W} \mid \pi_W \tag{18.6}$$

where \widetilde{W} denotes all the other variables except the parents and descendants of W .

Proof. First, we assume that $W \perp\!\!\!\perp \widetilde{W} \mid \pi_W$ and want to show that $p(x) = \prod_{j=1}^d p(x_j \mid \pi_{x_j})$, where $p(x)$ is the density function.

Without loss of generality, we let X_1, X_2, \dots, X_d be a topological ordering of the variables in the graph G . The following result follows from the chain rule:

$$p(x) = \prod_{j=1}^d p(x_j | x_1, \dots, x_{j-1}). \quad (18.7)$$

From (18.6), it is easy to see that $p(x_j | x_1, \dots, x_{j-1}) = p(x_j | \pi_{x_j})$. This proves one direction. The other direction is left as an exercise (see Exercise 1). \square

Write $\mathcal{I}(G)$ to denote the set of conditional independence relations corresponding to the graph G . Also write $\mathcal{I}(P)$ to denote the set of conditional independence relations corresponding to the distribution P . Then we can restate Theorem 18.5 by saying that $P \in \mathcal{M}(G)$ if and only if $\mathcal{I}(G) \subset \mathcal{I}(P)$. Say that P is *faithful* to G if $\mathcal{I}(G) = \mathcal{I}(P)$.

18.8 Example. Let $G = (V, E)$ where $V = (X, Y)$ and $E = \{(X, Y)\}$. The graph has one edge from node X to node Y , denoted as $X \rightarrow Y$. Then $\mathcal{I}(G) = \emptyset$. Suppose P has probability function $p(x)$ and that $p(x, y) = p(x)p(y)$. Then $\mathcal{I}(P) = \{X \perp\!\!\!\perp Y\}$. Hence $\mathcal{I}(G) \subset \mathcal{I}(P)$. Therefore P is Markov to G but is not faithful to G . \square

18.9 Example. Consider the DAG in Figure 18.6, the Markov condition implies that $X \perp\!\!\!\perp Y$ and $W \perp\!\!\!\perp \{X, Y\} | Z$. \square

18.10 Example. Consider the DAG in Figure 18.7. In this case the probability function

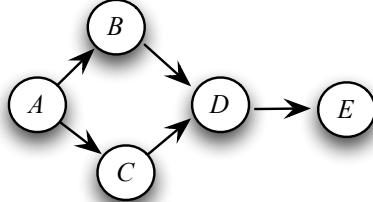


Figure 18.7. Yet another DAG.

must factor like $p(a, b, c, d, e) = p(a)p(b | a)p(c | a)p(d | b, c)p(e | d)$.

The Markov condition implies the following conditional independence relations:

$$D \perp\!\!\!\perp A | \{B, C\}, \quad E \perp\!\!\!\perp \{A, B, C\} | D \quad \text{and} \quad B \perp\!\!\!\perp C | A$$

\square

18.11 Example. Let G be a chain graph denoted as in Figure 18.8. We then have $p(x) = p(x_0)p(x_1 | x_0)p(x_2 | x_1) \dots$. The distribution of each variable depends only on its immediate predecessor. We say that P is a *Markov chain*. \square

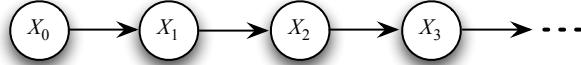


Figure 18.8. A chain graph.

18.12 Example. A *hidden Markov model (HMM)* involves two sets of variables X_1, X_2, \dots and Y_1, Y_2, \dots . The X_i 's form a Markov chain but they are unobserved. The Y_i 's are observed and the distribution of Y_i depends only on X_i . See Figure 18.9, in which we use gray variables to indicate the fact that Y_i 's are observed. HMMs are used in genetics, speech recognition and many other applications. \square

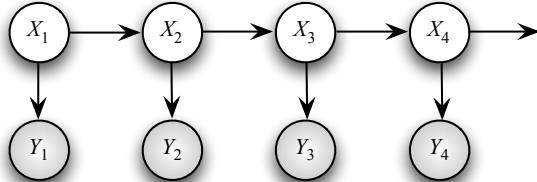


Figure 18.9. A hidden Markov model. The gray variables indicate that they are observed.

18.13 Example. Some statistical models can naturally be written in layers and are called *hierarchical models* or *random effects models*. For example, suppose we sample k counties and then we sample n_i people in the i -th county. We count the number Y_i that test positive for some disease. Then $Y_i \sim \text{Binomial}(n_i, \theta_i)$. The θ_i 's can also be regarded as random variables sampled from some distribution $p(\theta; \psi)$. For example, we might have $\theta_i \sim \text{Beta}(\alpha, \beta)$ so that $\psi = (\alpha, \beta)$. The model can be written as

$$\begin{aligned}\theta_i &\sim \text{Beta}(\alpha, \beta) \\ Y_i | \theta_i &\sim \text{Binomial}(n_i, \theta_i).\end{aligned}$$

Figure 18.10 shows a DAG representing this model. \square

18.4 More Independence Relations

The Markov condition allows us to list some independence relations implied by a DAG. These relations might imply other independence relations. Let's consider the DAG in Figure 18.11. The Markov condition implies:

$$\begin{aligned}X_1 \perp\!\!\!\perp X_2, \quad X_2 \perp\!\!\!\perp \{X_1, X_4\}, \quad X_3 \perp\!\!\!\perp X_4 \mid \{X_1, X_2\}, \\ X_4 \perp\!\!\!\perp \{X_2, X_3\} \mid X_1, \quad X_5 \perp\!\!\!\perp \{X_1, X_2\} \mid \{X_3, X_4\}\end{aligned}$$

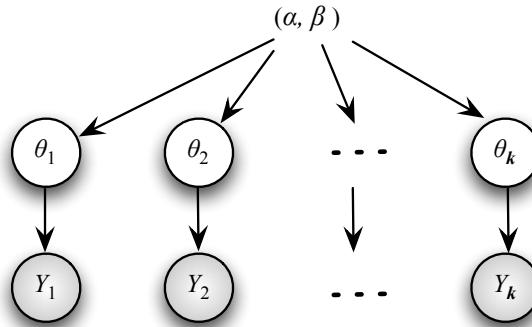


Figure 18.10. A hierarchical model. All Y_i 's are represented by gray nodes, indicating the fact that they are observed.

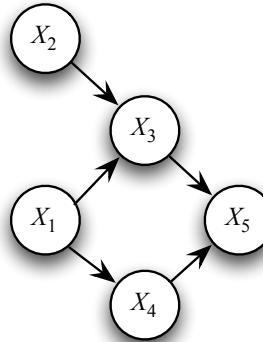


Figure 18.11. And yet another DAG.

It turns out (but it is not obvious) that these conditions imply that

$$\{X_4, X_5\} \perp\!\!\!\perp X_2 \mid \{X_1, X_3\}.$$

How do we find these extra conditional independence relations? The answer is “*d-separation*,” which is short for “*directed separation*.” The notion of d-separation can be summarized by three rules. Consider the four DAGs in Figure 18.12. The first three DAGs in Figure 18.12 have no colliders. The DAG in Figure 18.12 (d) has a collider. The DAG in Figure 18.4 has a collider with a descendant.

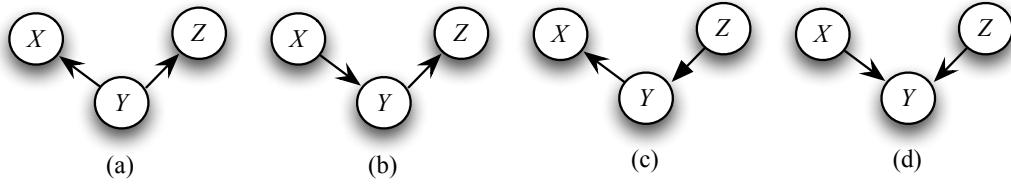


Figure 18.12. The first three DAGs have no colliders. The fourth DAG has a collider at Y .

The Rules of d-Separation

Consider the DAGs in Figures 18.12 and 18.4.

1. When Y is not a collider, X and Z are *d-connected*, but they are *d-separated* given Y .
2. If X and Z collide at Y , then X and Z are *d-separated*, but they are *d-connected* given Y .
3. Conditioning on the descendant of a collider has the same effect as conditioning on the collider. Thus in Figure 18.4, X and Z are *d-separated* but they are *d-connected* given W .

Here is a more formal definition of d-separation. Let X and Y be distinct vertices and let W be a set of vertices not containing X or Y . Then X and Y are *d-separated* given W if there exists no undirected path U between X and Y such that (i) every collider on U has a descendant in W , and (ii) no other vertex on U is in W . If A , B , and W are distinct sets of vertices and A and B are not empty, then A and B are d-separated given W if for every $X \in A$ and $Y \in B$, X and Y are d-separated given W . The sets of vertices that are not d-separated are said to be d-connected.

18.14 Example. Consider the DAG in Figure 18.13.

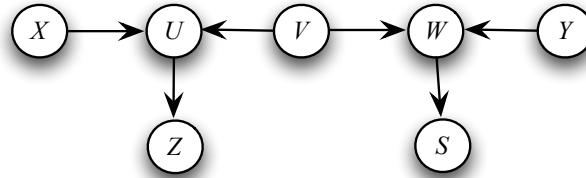


Figure 18.13. Example for d-separation.

From the d-separation rules we conclude that: (i) X and Y are d-separated (given the empty set); (ii) X and Y are d-connected given $\{Z, S\}$; (iii) X and Y are d-separated given $\{Z, S, V\}$. \square

The following theorem, due to Verma and Pearl (1988), provides the probabilistic implications of d-separation.

18.15 Theorem. (Verma and Pearl (1988)) *If sets A and B are d-separated by C in a DAG G , then A is independent of B conditional on C in every distribution compatible with G . Conversely, if A and B are not d-separated by C in a DAG G , then A and B are dependent conditional on C in at least one distribution compatible with G .*

Proof Idea. Assuming A and B are d-separated by C , the proof uses the algebraic properties of conditional independence. For the converse, the proof constructs a distribution P such that A and B are correlated conditioned on C . Such a construction is possible since A and B are not d-separated by C . From the definition of d-separability, there are several cases to consider, depending on whether a collider appears between A and B . In each case, conditional probability distributions can be constructed so that the correlations can be propagated from A to B . \square

18.16 Example. Consider the DAG in Figure 18.5. In this example, overweight and smoking are marginally independent but they are dependent given heart disease. \square

Graphs that look different may actually imply the same independence relations. If G is a DAG, we let $\mathcal{I}(G)$ denote all the independence statements implied by G . Two DAGs G_1 and G_2 for the same variable set V are *Markov equivalent* if $\mathcal{I}(G_1) = \mathcal{I}(G_2)$. All DAGs can be partitioned into equivalence classes so that DAGs within each class are Markov equivalent. Given a DAG G , let $\text{skeleton}(G)$ denote the undirected graph obtained by replacing the arrows with undirected edges.

18.17 Theorem. (Verma and Pearl (1990)) *Two DAGs G_1 and G_2 are Markov equivalent if and only if (i) $\text{skeleton}(G_1) = \text{skeleton}(G_2)$ and (ii) G_1 and G_2 have the same unshielded colliders (i.e. any two nodes pointing to the same collider are not connected).*

Proof. See Exercise 3. \square

18.18 Example. According to Theorem 18.17, the first three DAGs in Figure 18.12 are Markov equivalent. The DAG in Figure 18.12 (d) is not Markov equivalent to the other three. \square

18.5 Gaussian DAGs

We now represent the multivariate Gaussian distribution using a directed graphical model. This representation is based on the *linear Gaussian model*, which is defined in the following

18.19 Definition. (Linear Gaussian Model) Let Y be a continuous variable in a DAG with parents X_1, \dots, X_k . We say that Y has a linear Gaussian model of its parents if there are parameters $\beta_0, \beta_1, \dots, \beta_k$ and σ^2 such that

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + \epsilon, \quad \epsilon \sim N(0, \sigma^2). \quad (18.20)$$

18.21 Definition. (Gaussian Directed Graphical Models) A directed graphical model is called a Gaussian directed graphical model (or Gaussian Bayesian network) if all the variables are continuous and every variable and its parents follow a linear Gaussian model.

In a Gaussian Bayesian network, each variable X_j is modeled as a linear function of its parents plus normally distributed random noise. One important result is that there is one-to-one correspondence between a multivariate Gaussian distribution and a Gaussian Bayesian network. The next theorem shows that Gaussian Bayesian network defines a joint multivariate Gaussian distribution.

18.22 Theorem. We assume that Y has a linear Gaussian model of its parents X_1, \dots, X_k : $Y = \beta_0 + \sum_{j=1}^k \beta_j X_j + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. We also assume that $X = (X_1, \dots, X_k)$ are jointly Gaussian with distribution $N(\mu, \Sigma)$. Then the joint distribution of (Y, X_1, \dots, X_k) is a Gaussian distribution with $\text{Cov}(Y, X_i) = \sum_{j=1}^k \beta_j \Sigma_{ij}$. Moreover, we have $\mathbb{E}(Y) = \beta_0 + \beta^T \mu$ and $\text{Var}(Y) = \sigma^2 + \beta^T \Sigma \beta$, where $\beta = (\beta_1, \dots, \beta_k)$.

Proof. Since the marginal distribution of X and the conditional distribution of Y given X are both Gaussians, the joint distribution of (Y, X) is Gaussian with the covariance:

$$\text{Cov}(Y, X_i) = \text{Cov}\left(\beta_0 + \sum_{j=1}^k \beta_j X_j, X_i\right) = \sum_{j=1}^k \beta_j \text{Cov}(X_j, X_i) = \sum_{j=1}^k \beta_j \Sigma_{ij}. \quad (18.23)$$

Also, since Y is the summation of $k+1$ Gaussian variables, the marginal distribution of Y is Gaussian with the desired mean and variance. \square

The converse of this theorem is also true, which states that any multivariate Gaussian distribution can be converted to a Gaussian Bayesian network. We first state a simple lemma, the proof of which follows from straightforward algebraic manipulation.

18.24 Lemma. Let (Y, X) be a multivariate Gaussian distribution:

$$\begin{pmatrix} Y \\ X \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_Y \\ \mu_X \end{pmatrix}, \begin{pmatrix} \Sigma_{YY} & \Sigma_{YX} \\ \Sigma_{XY} & \Sigma_{XX} \end{pmatrix}\right). \quad (18.25)$$

Then $Y | X \sim N(\beta_0 + \beta^T X, \sigma^2)$ where

$$\beta_0 = \mu_Y - \Sigma_{YX} \Sigma_{XX}^{-1} \mu_X, \quad \beta = \Sigma_{XX}^{-1} \Sigma_{YX}, \quad \text{and } \sigma^2 = \Sigma_{YY} - \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}.$$

18.26 Theorem. Let P be the joint distribution of d -dimensional multivariate Gaussian random vector $X = (X_1, \dots, X_d)$. For any ordering of the variables $X_{\tau(1)}, \dots, X_{\tau(d)}$, we can construct a DAG G such that P is Markov to G , and where $X_{\tau(i)}$ is a linear Gaussian model of its parents $\pi(X_{\tau(i)}) \subset \{X_{\tau(1)}, \dots, X_{\tau(i-1)}\}$ for all i .

Proof. See Exercise 4. \square

Next, we study the conditional independence properties of a Gaussian model. We start with the definition of *partial correlation*, which measures the degree of association between two random variables, with the effect of a set of controlling random variables removed.

18.27 Definition. The partial correlation $\rho_{X,Y|Z}$ between two variables X and Y given another set of variables $Z = (Z_1, Z_2, \dots, Z_k)$ is the correlation between the residuals resulting from regressing X on Z and Y on Z . More formally,

$$\rho_{X,Y|Z} = \frac{\mathbb{E}(XY|Z) - \mathbb{E}(X|Z)\mathbb{E}(Y|Z)}{\sqrt{\text{Var}(X|Z)}\sqrt{\text{Var}(Y|Z)}}. \quad (18.28)$$

Conditional independencies can be inferred from partial correlations in a multivariate Gaussian distribution. This result is summarized in the next theorem, which follows from the elementary properties of the multivariate Gaussian (Lauritzen, 1996a).

18.29 Theorem. Let X be a Gaussian random vector $X = (X_1, \dots, X_d)$. For $i \neq j \in \{1, \dots, d\}$, $K \subset \{1, \dots, d\} \setminus \{i, j\}$, denote by $\rho_{i,j|K}$ the partial correlation between X_i and X_j given $\{X_k : k \in K\}$. Then $\rho_{i,j|K} = 0$ if and only if X_i and X_j are conditionally independent given $\{X_k : k \in K\}$.

We can thus obtain estimates of conditional independencies for Gaussian DAGs by calculating sample partial correlations. These can be computed via regression, computing components of the inverse of the sample covariance, or recursively using the following proposition.

18.30 Proposition. Let $X = (X_1, \dots, X_d)$ be multivariate Gaussian. For $i \neq j \in \{1, \dots, d\}$, $K \subset \{1, \dots, d\} \setminus \{i, j\}$, let $\rho_{i,j|K}$ be the partial correlation between X_i and X_j given $\{X_k : k \in K\}$. Then for any $h \in K$,

$$\rho_{i,j|K} = \frac{\rho_{i,j|K \setminus h} - \rho_{i,h|K \setminus h} \cdot \rho_{j,h|K \setminus h}}{\sqrt{1 - \rho_{i,h|K \setminus h}^2} \sqrt{1 - \rho_{j,h|K \setminus h}^2}}. \quad (18.31)$$

Proof. See Exercise 5. \square

In the above proposition, when A is the empty set, the partial correlation $\rho_{i,j|A}$ reduces to the Pearson correlation between the random variables X_i and X_j . This enables calculation of higher order partial correlations in a recursive way.

18.6 Exact Inference

For *inference in directed graphical models*, some of the variables in a graph are set to certain values due to evidence, and we wish to compute the posterior distribution of some other variables. This requires calculating the probability distribution on a subset of variables by marginalizing over the joint. One thing to note is that inference in this section is just calculating marginal and conditional probabilities, which is different from the general term statistical inference used in this book.

Even a probability distribution is given, calculating marginal or conditional distributions can be costly because it requires summing over an exponential number of joint probability combinations. The first efficient algorithm proposed for probabilistic inference in DAGs used *message-passing* architecture and were limited to trees (Pearl, 1982). The main idea is to view each variable as a simple processor and reduce probabilistic inference to asynchronous local message passing among different nodes until equilibrium is achieved. In this section, we mainly focus on *exact inference* methods, and in the later computing chapters we will consider different *approximate inference* algorithms.

18.6.1 Belief Propagation (Sum-Product Algorithm) on Polytrees

We now introduce a local message-passing type algorithm named *belief propagation* (Pearl, 1988; Lauritzen and Spiegelhalter, 1988). For simplicity, we assume all the variables are discrete, therefore marginalization only requires summations. This assumption is not restrictive since for continuous variables we only need to replace summation with integration.

We focus on conducting exact inference on *polytrees*. In graph theory, a polytree is a directed graph with at most one undirected path between any two nodes. In other words, a polytree is a DAG for which there are no undirected cycles. One example of polytree is provided in Figure 18.14. To introduce belief propagation algorithm for polytrees, we consider an inference problem that takes the form $\mathbb{P}(X = x | E = e)$ where X is a *query node* and E is any subset of observed nodes (or *evidence nodes*) who have been set to certain values.

Without loss of generality, we assume $E = E_X^+ \cup E_X^-$ where E_X^+ is a subset of the ancestor nodes of X and E_X^- is a subset of the descendant nodes of X (Both E_X^+ and E_X^- may also be empty). For belief propagation, we treat each node as a processor that receives messages from its neighbors and pass them along after some local calculation. Let X has n children Y_1, \dots, Y_n and k parents Z_1, \dots, Z_k . The node X receives and processes two types of messages from its neighbors. First, we use $m_{Y_j}^-(X)$ to represent the message sent from the child Y_j to X . Assuming X has n children, we have n such messages:

$$m_{Y_1}^-(X), \dots, m_{Y_n}^-(X). \quad (18.32)$$

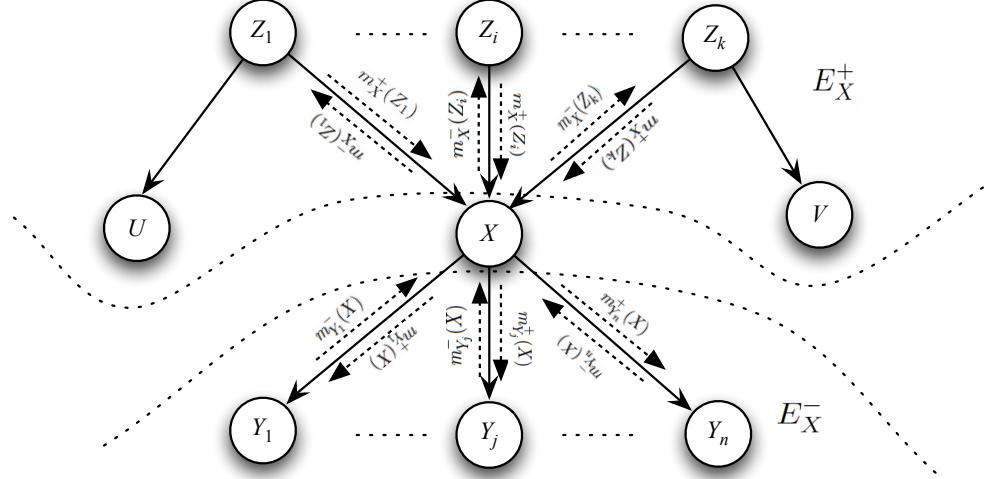


Figure 18.14. A polytree that is used to illustrate the belief propagation algorithm. In a polytree, a node may have several children and parents, but there can be at most one undirected path that connects any two nodes.

More detailed forms of $m_{Y_j}^-(X)$ will be provided in later paragraphs. We also denote $m_{Y_j}^+(X)$ to be the message sent from X to one of its children Y_j . For all the n children of X , we have the following messages:

$$m_{Y_1}^+(X), \dots, m_{Y_n}^+(X). \quad (18.33)$$

Assuming the node X has exactly k parents, we define $m_X^-(Z_1), \dots, m_X^-(Z_k)$ to be the messages sent from X to its parents Z_1, \dots, Z_k . Similarly, we define $m_X^+(Z_1), \dots, m_X^+(Z_k)$ to be the messages sent from Z_1, \dots, Z_k to X . For more details, see Figure 18.14.

Let $E_X^+ = e_X^+$ and $E_X^- = e_X^-$. We want to evaluate

$$p(x | e_X^+, e_X^-) \equiv \mathbb{P}(X = x | E_X^+ = e_X^+, E_X^- = e_X^-). \quad (18.34)$$

Since X d-separates E_X^+ and E_X^- , we have that $p(e_X^+, e_X^- | x) = p(e_X^+ | x)p(e_X^- | x)$.

We further define $E_X^+ = E_{X,Z_1}^+ \cup \dots \cup E_{X,Z_k}^+$ where E_{X,Z_i}^+ is a subset of E_X^+ that are also ancestors of Z_i . Similarly, we define $E_X^- = E_{X,Y_1}^- \cup \dots \cup E_{X,Y_n}^-$ where E_{X,Y_j}^- is a subset of E_X^- that are also descendants of Y_j . We define $m^+(x) \equiv p(x | e_X^+)$ to be the propagated $E_X^+ = e_X^+$ that X receives from its parents and passes on to its children, and $m^-(x) \equiv p(e_X^- | x)$ to be the propagated $E_X^- = e_X^-$ that X receives from its children and

passes on to its parents. We then have

$$p(x | e_X^+, e_X^-) = \frac{p(e_X^+, e_X^- | x)p(x)}{p(e_X^+, e_X^-)} = \frac{p(e_X^+ | x)p(e_X^- | x)p(x)}{p(e_X^+, e_X^-)} \quad (18.35)$$

$$= \frac{p(x | e_X^+)p(e_X^+)p(e_X^- | x)p(x)}{p(x)p(e_X^+, e_X^-)} \quad (18.36)$$

$$= \alpha p(x | e_X^+)p(e_X^- | x) \quad (18.37)$$

$$= \alpha m^+(x)m^-(x) \quad (18.38)$$

where the normalizing constant $\alpha = p(e_X^+)/p(e_X^+, e_X^-)$ does not depend on x .

We now explain how to evaluate $m^+(x)$ and $m^-(x)$ in a recursive way. Similarly to the definition of $m^+(x)$ and $m^-(x)$, we define the messages

$$m_{Y_j}^+(x) = p(x | e_{X,Y_j}^+) \text{ and } m_{Y_j}^-(x) = p(e_{X,Y_j}^- | x), \quad (18.39)$$

$$m_X^+(z_i) = p(z_i | e_{X,Z_i}^+) \text{ and } m_X^-(z_i) = p(e_{X,Z_i}^- | z_i). \quad (18.40)$$

We then have

$$m^-(x) = p(e_X^- | x) = p(e_{X,Y_1}^-, \dots, e_{X,Y_n}^- | x) = \prod_{j=1}^n p(e_{X,Y_j}^- | x) = \prod_{j=1}^n m_{Y_j}^-(x), \quad (18.41)$$

where we have utilized the fact that X d-separates Y_1, \dots, Y_n .

Similarly, we have

$$m^+(x) = p(x | e_X^+) = p(x | e_{X,Z_1}^+, \dots, e_{X,Z_k}^+) \quad (18.42)$$

$$= \sum_{z_1} \sum_{z_2} \cdots \sum_{z_k} p(x | z_1, z_2, \dots, z_k) p(z_1, \dots, z_k | e_{X,Z_1}^+, \dots, e_{X,Z_k}^+) \quad (18.43)$$

$$= \sum_{z_1} \sum_{z_2} \cdots \sum_{z_k} p(x | z_1, z_2, \dots, z_k) \prod_{i=1}^k p(z_i | e_{X,Z_i}^+) \quad (18.44)$$

$$= \sum_{z_1} \sum_{z_2} \cdots \sum_{z_k} p(x | z_1, z_2, \dots, z_k) \prod_{i=1}^k m_X^+(z_i). \quad (18.45)$$

Therefore, we see that given all the messages passed from the node X 's parents and children, we can evaluate $p(x | e_X^+, e_X^-)$ easily. The remaining question is how can we efficiently collect these messages. This requires us to calculate the messages a node send to its children and parents. Without loss of generality, we only need to explain how to calculate the messages that X send to its parents and children:

$$m_X^-(Z_1), \dots, m_X^-(Z_k), \text{ and } m_{Y_1}^+(X), \dots, m_{Y_n}^+(X).$$

Note that what node Y_j receives from X includes both information that X gets from its parents Z_1, \dots, Z_k (i.e. E_X^+) and also from its other children $Y_1, \dots, Y_{j-1}, Y_{j+1}, \dots, Y_n$

(i.e. $\{E_X^- \setminus E_{X,Y_j}^-\}$). Let

$$\beta = \frac{1}{p(\{e_X^- \setminus e_{X,Y_j}^-\} | e_X^+)}.$$

To evaluate $m_{Y_j}^+(X)$, we have

$$m_{Y_j}^+(x) = p(x | e_{X,Y_j}^+) = p\left(x | e_X^+, \{e_X^- \setminus e_{X,Y_j}^-\}\right) \quad (18.46)$$

$$= \frac{p\left(\{e_X^- \setminus e_{X,Y_j}^-\} | x, e_X^+\right) p(x | e_X^+)}{p\left(\{e_X^- \setminus e_{X,Y_j}^-\} | e_X^+\right)} \quad (18.47)$$

$$= \beta \cdot p\left(\{e_X^- \setminus e_{X,Y_j}^-\} | x, e_X^+\right) p(x | e_X^+) \quad (18.48)$$

$$= \beta \prod_{\ell \neq j} p\left(e_{X,Y_\ell}^- | x, e_X^+\right) p(x | e_X^+) \quad (18.49)$$

$$= \beta \prod_{\ell \neq j} p\left(e_{X,Y_\ell}^- | x\right) p(x | e_X^+) \quad (18.50)$$

$$= \beta \prod_{\ell \neq j} m_{Y_\ell}^-(x) m^+(x). \quad (18.51)$$

Where have used the fact that X d-separates Y_1, \dots, Y_n .

We then show how to evaluate $m_X^-(Z_i)$. Note that the message $m_X^-(Z_i)$ that X passes on to one of its parents Z_i includes not only the messages X gets from its children (i.e. E_X^-) but also the messages X receives from its other parents (i.e. $\{E_X^- \setminus E_{X,Z_i}^-\}$). Let

$$\gamma = p\left(\{e_X^+ \setminus e_{X,Z_i}^+\}\right). \quad (18.52)$$

We then have

$$m_X^-(z_i) = p(e_{X,Z_i}^- | z_i) = \sum_x \sum_{\{z_\ell\}_{\ell \neq i}} p(e_{X,Z_i}^-, x, \{z_\ell\}_{\ell \neq i} | z_i) \quad (18.53)$$

$$= \sum_x \sum_{\{z_\ell\}_{\ell \neq i}} p(e_X^-, \{e_X^+ \setminus e_{X,Z_i}^+\}, x, \{z_\ell\}_{\ell \neq i} | z_i) \quad (18.54)$$

$$= \sum_x \sum_{\{z_\ell\}_{\ell \neq i}} p(e_X^-, \{e_X^+ \setminus e_{X,Z_i}^+\} | x, \{z_\ell\}_{\ell \neq i}, z_i) p(x, \{z_\ell\}_{\ell \neq i} | z_i) \quad (18.55)$$

$$= \sum_x \sum_{\{z_\ell\}_{\ell \neq i}} p(e_X^- | x) p(\{e_X^+ \setminus e_{X,Z_i}^+\} | \{z_\ell\}_{\ell \neq i}) p(x, \{z_\ell\}_{\ell \neq i} | z_i) \quad (18.56)$$

$$= \sum_x \sum_{\{z_\ell\}_{\ell \neq i}} p(e_X^- | x) \frac{p(\{z_\ell\}_{\ell \neq i} | \{e_X^+ \setminus e_{X,Z_i}^+\}) \cdot \gamma}{p(\{z_\ell\}_{\ell \neq i})} p(x, \{z_\ell\}_{\ell \neq i} | z_i) \quad (18.57)$$

$$= \gamma \sum_x \sum_{\{z_\ell\}_{\ell \neq i}} p(e_X^- | x) p(\{z_\ell\}_{\ell \neq i} | \{e_X^+ \setminus e_{X,Z_i}^+\}) p(x | \{z_\ell\}_{\ell \neq i}, z_i) \quad (18.58)$$

$$= \gamma \sum_x \sum_{\{z_\ell\}_{\ell \neq i}} m^-(x) \left(\prod_{\ell \neq i} m_X^+(z_\ell) \right) p(x | z_1, \dots, z_k) \quad (18.59)$$

$$= \gamma \sum_x m^-(x) \sum_{\{z_\ell\}_{\ell \neq i}} p(x | z_1, \dots, z_k) \prod_{\ell \neq i} m_X^+(z_\ell). \quad (18.60)$$

Given the above recursive relationship of the message-passing algorithm, the only thing left is to determine the initial values of the leaf nodes, root nodes, and evidence nodes. Let X be a node with parents Z_1, \dots, Z_k and children Y_1, \dots, Y_n (If X is a leaf node, then there is no children; If X is a root node, then there is no parents). It is easy to see that if X is initialized to be a certain evidence value e , then

$$m_X^-(Z_1) = \dots = m_X^-(Z_n) = m_{Y_1}^+(X) = \dots = m_{Y_n}^+(X) = 1. \quad (18.61)$$

If X is an un-initialized leaf node with parents Z_1, \dots, Z_k , we have $m_X^-(Z_1) = \dots = m_X^-(Z_k) = 1$. If X is an un-initialized root node with children Y_1, \dots, Y_n , we have $m_{Y_1}^+(X) = \dots = m_{Y_n}^+(X) = \mathbb{P}(X) = p(x)$.

The belief propagation algorithm conducts exact inference for polytrees. However, if there is a cycle in the underlying undirected graph, the belief propagation algorithm will no longer work. The main reason is that when cycles exist, a node X is no longer guaranteed to d-separate its ancestors and descendants. Therefore, the messages can be propagated through multiple paths. To apply belief propagation, we need to convert the graph into a polytree T . Each node of T may correspond to a set of original variables. This is related to the *junction tree* algorithms, which can be viewed as belief propagation on a modified graph guaranteed to be a polytree. The basic idea is to eliminate cycles by clustering them into single nodes. More details about junction trees will be introduced in the next chapter

on undirected graphical models. Since the messages in the belief propagation algorithm takes the form of summing over many product terms. The belief propagation is also called the *sum-product* algorithm.

18.6.2 Max-Product and Max-Sum Algorithms

A similar algorithm is commonly referred to as *max-product*, also known as *max-sum* algorithm, which solves a related problem of maximizing a probability, or finding the most probable explanation for certain observations. Instead of attempting to calculate the marginal, the goal now is to find a joint configuration of $X = \hat{x}$ which has *maximum a posterior probability* (the *MAP* estimate):

$$\hat{x} = \arg \max_x \mathbb{P}(X = x | E = e), \quad (18.62)$$

where X is a *query node* which could possibly represents a set of random variables. E is the evidence node as before. An algorithm that solves this problem is nearly identical to belief propagation, with sums replaced by maxima in messages. The resulting procedure is called the *max-product* algorithm. Note that to avoid potential underflow, the max-product algorithm can equivalently be written in terms of log-probabilities, in which case one obtains the *max-sum* algorithm. Once again we see that in a polytree, exact MAP inference can be performed in two sweeps using the max-product algorithm. The algorithm is in fact an example of dynamic programming. The max-product or max-sum algorithm, when applied to hidden Markov models (HMMs), is known as the *Viterbi algorithm*.

18.7 Approximate Inference

The graphical models encountered in applications may have large cliques or long loops, which make exact inference intractable. In this setting we must conduct *approximate inference*. There are two popular ways for approximate inference: (i) *variational methods* and (ii) *sampling* methods. The former class of methods are deterministic and the latter class of methods are stochastic. We defer the discussions of these two families of approximation inference algorithms to later computing chapters. Another way to conduct approximate inference is to directly apply the belief propagation algorithm without worrying about the loops. Such a method is known as *loopy belief propagation* (Frey and MacKay, 1997). This algorithm can be effective on certain applications, though its convergence is not guaranteed.

18.8 Parameter Estimation

Two estimation questions arise in the context of DAGs. First, given a DAG G and data x_1, \dots, x_n from a distribution $p(x)$ consistent with G , how do we estimate $p(x)$? Second, given data x_1, \dots, x_n how do we estimate G ? The first question is a standard *parameter estimation* problem. The second question is a *structure learning* problem and is similar in

approaches and terminology to model selection procedures for classical statistical models. In this section we only discuss parameter estimation with pre-fixed DAG G . For parameter estimation, one important distinction is whether all the variables are observed, or whether some of them are hidden. We discuss these two cases separately.

18.8.1 Parameter Estimation from Fully Observed Data

Let G be a DAG with vertices $V = (X_1, \dots, X_d)$. Once G is given, the task of estimating the parameters of the joint distribution can be greatly simplified by the application of the Markov property. Suppose we use a parametric model $p(x_j | \pi_{x_j}; \theta_j)$ for each conditional density in (18.2), where π_{x_j} is the set of parent nodes of X_j . Let $\theta = (\theta_1, \dots, \theta_d)$ be the set of parameters, the joint distribution in (18.2) can be written as

$$p(x; \theta) = \prod_{j=1}^d p(x_j | \pi_{x_j}; \theta_j). \quad (18.63)$$

Given n data points $\{x_1, \dots, x_n\}$, the likelihood function is

$$\mathcal{L}(\theta) = \prod_{i=1}^n p(x_i; \theta) = \prod_{i=1}^n \prod_{j=1}^d p(x_{ij} | \pi_{x_j}; \theta_j), \quad (18.64)$$

where x_{ij} is the value of X_j for the i^{th} data point and θ_j are the parameters for the j^{th} conditional density. We can then estimate the parameters by maximum likelihood. It is easy to see that the log-likelihood decomposes according to the graph structure:

$$\ell(\theta) = \log \mathcal{L}(\theta) = \sum_{j=1}^d \log \left(\prod_{i=1}^n p(x_{ij} | \pi_{x_j}; \theta_j) \right) \equiv \sum_{j=1}^d \log \mathcal{L}_j(\theta_j) = \sum_{j=1}^d \ell_j(\theta_j), \quad (18.65)$$

where $\ell_j(\theta_j) = \sum_{i=1}^n \log p(x_{ij} | \pi_{x_j}; \theta_j)$ is a localized conditional likelihood for θ_j . Therefore we can maximize the contribution to the log-likelihood of each node independently. (It is straightforward to extend this to the shared parameter paradigms.)

When there is not enough information from the data points, we could also regularize the log-likelihood to avoid overfitting:

$$\hat{\theta} = \arg \max_{\theta} \{\ell(\theta) - \text{Pen}(\theta)\}, \quad (18.66)$$

where $\text{Pen}(\theta) \geq 0$ is some penalty term of θ .

18.8.2 Parameter Estimation with Hidden Variables

In many applications, observed data may not include the values of some of the variables in the DAG. We refer to these variables as hidden variables. If Z denotes the hidden

variables, the log-likelihood can be written as

$$\ell(\theta) = \sum_{i=1}^n \log p(x_i; \theta) = \sum_{i=1}^n \log \int_{z_i} p(x_i, z_i; \theta) dz_i. \quad (18.67)$$

With hidden variables, the log-likelihood is no longer decomposable as in (18.65), and maximizing the log-likelihood in (18.67) is often difficult. This can be approached using the EM algorithm, which is discussed in later chapters.

18.68 Example. Consider the graphical models in Figure 18.15. The DAG has four nodes. Each node corresponds to one univariate random variable. We consider two settings: (a) all the four variables are fully observable, with data $\{(x_i, y_i, z_i, w_i)\}_{i=1}^n$; (b) only three variables X, Z, W are observable, with data $\{(x_i, z_i, w_i)\}_{i=1}^n$. Given the DAG topology, we

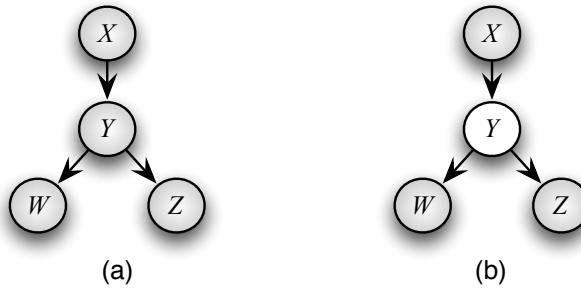


Figure 18.15. (a) a DAG where all the four nodes are observable; (b) a DAG where only three nodes are observable and one node Y is hidden. A node is gray-colored if it is observable.

know that the joint density has the decomposition $p(x, y, z, w) = p(w | y)p(z | y)p(y | x)p(x)$. We parametrize the conditional distributions as the following:

$$W | Y = y \sim N(\mu_1, 1)I(y = 1) + N(\mu_0, 1)I(y = 0) \quad (18.69)$$

$$Z | Y = y \sim N(\mu_0, 1)I(y = 1) + N(\mu_1, 1)I(y = 0) \quad (18.70)$$

$$Y | X = x \sim \text{Bernoulli} \left(\frac{1}{1 + \exp(-\beta_0 - \beta_1 x)} \right) \quad (18.71)$$

$$X \sim N(\mu_2, \sigma^2). \quad (18.72)$$

From the above parameterization, we see that the conditional distributions $p(w | y)$ and $p(z | y)$ share the same set of parameters μ_0 and μ_1 . Let $\theta = (\mu_0, \mu_1, \beta_0, \beta_1, \mu_2, \sigma)$ and $\phi(\cdot)$ be the standard Gaussian density function. When all the four variables are observable, the joint log-likelihood of θ has the following decomposition

$$\ell(\theta) = \ell(\mu_0, \mu_1) + \ell(\beta_0, \beta_1) + \ell(\mu_2, \sigma^2) + \text{constant}, \quad (18.73)$$

where $\ell(\mu_0, \mu_1) = -\frac{1}{2} \sum_{i=1}^n I(y_i = 1) \cdot [(w_i - \mu_1)^2 + (z_i - \mu_0)^2] - \frac{1}{2} \sum_{i=1}^n I(y_i = 0) \cdot [(w_i - \mu_0)^2 + (z_i - \mu_1)^2]$, and $\ell(\beta_0, \beta_1) = -\sum_{i=1}^n I(y_i = 1) \log (1 + \exp(\beta_0 + \beta_1 x_i)) -$

$$\sum_{i=1}^n I(y_i = 0) \log(1 + \exp(-\beta_0 - \beta_1 x_i)). \text{ We also have } \ell(\mu_2, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu_2)^2.$$

It is easy to see that the maximum likelihood estimates take the form

$$\hat{\mu}_0 = \frac{1}{2n} \sum_{i=1}^n [I(y_i = 0) \cdot w_i + I(y_i = 1) \cdot z_i] \quad (18.74)$$

$$\hat{\mu}_1 = \frac{1}{2n} \sum_{i=1}^n [I(y_i = 1) \cdot w_i + I(y_i = 0) \cdot z_i] \quad (18.75)$$

$$\hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{and} \quad \widehat{\sigma^2} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_2)^2. \quad (18.76)$$

The parameters β_0 and β_1 can also be easily estimated by solving a logistic regression using Y as output and X as input.

When Y is hidden (as in Figure 18.15 (b)), the log-likelihood no longer decomposes. Parameter estimation is more challenging and requires an iterative EM algorithm. \square

18.9 Structure Learning

Estimating a DAG from data is very challenging due to the enormous size of the space of DAGs (the number of possible DAGs is super-exponential in the number of nodes). Existing methods can be roughly divided into two categories: (i) *constraint-based* methods and *score-based* methods. Constraint-based methods use statistical tests to learn conditional independence relationships (called constraints in this setting) from the data and prune the graph-searching space using the obtained constraints. In contrast, score-based algorithms assign each candidate DAG a score reflecting its goodness of fit, which is then taken as an objective function to be optimized. In this section, we introduce a constraint-based method, named *PC algorithm* (after its authors, Peter and Clark, see Spirtes et al. (2000)), for estimating DAGs from observed data. Under certain conditions, a sample version of the PC algorithm has been shown to be consistent even for large graphs. In the following, we first describe the population PC algorithm (i.e. we assume the true distribution is given). We then explain if only n i.i.d. observations are obtained, how can we use these samples to estimate the corresponding population quantities of the PC algorithm.

18.9.1 Representing Equivalence Classes

Let $X = (X_1, \dots, X_d)$ be a d -dimensional random vector with distribution P . We assume that there exists a DAG G such that $\mathcal{I}(P) = \mathcal{I}(G)$. In another word, P is *faithful* to G . Without this assumption, the PC algorithm may fail.

With the faithfulness assumption, one obvious goal is to identify the DAG G from P . However, from Example 18.18, we know that there might exist another DAG G' which is Markov equivalent to G (i.e., G and G' are actually indistinguishable from P). Therefore, instead of identifying one single DAG G , we can only identify the whole Markov

equivalence class of G . From Theorem 18.17, we know that all DAGs in the equivalence class have the same skeleton and the set of unshielded colliders. Motivated by this theorem, the whole equivalence class can be compactly represented by the skeleton graph with unshielded colliders marked (All the edges are undirected except the edges corresponding to unshielded colliders). One such example is shown in Figure 18.16.

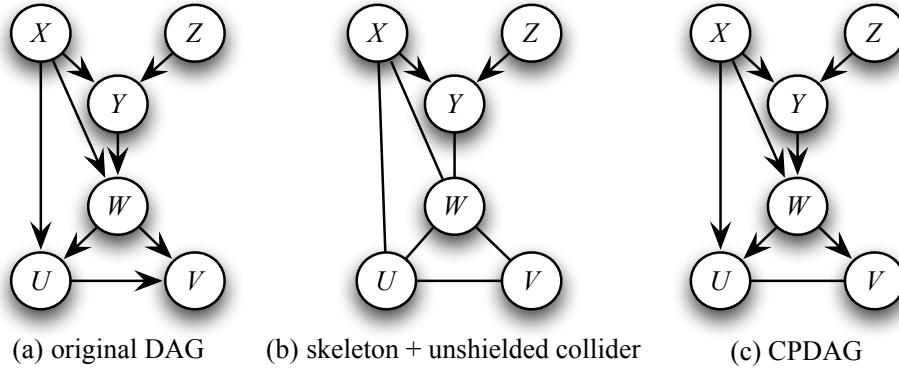


Figure 18.16. (a) The original DAG G ; (b) The skeleton of G with unshielded colliders; (c) The CPDAG corresponds to G . The figure shows that CPDAG can be more informative than just graph skeleton annotated with unshielded colliders.

Figure 18.16 (a) shows an original DAG G which has only one unshielded collider $X \rightarrow Y \leftarrow Z$. Figure 18.16 (b) shows the skeleton of G annotated with the unshielded collider. Every DAG that is Markov equivalent to G should have the same skeleton and the unshielded collider. Such a graph, containing both directed and undirected edges, is called *partially directed acyclic graph (PDAG)*. The definition of PDAG is:

18.77 Definition. (PDAG) A partially directed acyclic graph or PDAG is an acyclic graph containing both directed and undirected edges and one can not trace a cycle by following the direction of directed edges and any direction for undirected edges.

However, the representation in Figure 18.16 (b) is not compact at all. There are 6 undirected edges in the PDAG in Figure 18.16 (b). Since each edge have two possible directions, the potential number of DAGs represented by this graph is $2^6 = 64$. However, if we think more carefully, the directions of many undirected edges can in fact be determined based on the PDAG representing the skeleton and all the unshielded colliders. For example, the edge connecting Y and W must be $Y \rightarrow W$. Otherwise we will get another unshielded collider $Z \rightarrow Y \leftarrow W$, which contradicts the fact that the equivalence class has only one unshielded collider. Similarly, we get two more directed edges $W \rightarrow U$ and $W \rightarrow V$. Given the path $X \rightarrow Y \rightarrow W$, we immediately get the edge $X \rightarrow W$ since any element in the equivalence class must be a DAG (which does not contain loops). Similarly, given $X \rightarrow W \rightarrow U$, the edge connecting X and U must be $X \rightarrow U$. Therefore, the directions of many undirected edges in the skeleton can be determined using simple rules (e.g. we can not introduce new unshielded colliders or cycles.). Therefore the size of the

equivalence class can be greatly reduced. For the PDAG in Figure 18.16 (b), the only edge that we can not determine its directionality is $U \rightarrow V$, which lead to two potential DAGs.

We define a special PDAG, named *completed PDAG (CPDAG)*, to compactly represent an equivalent class:

18.78 Definition. (CPDAG) Let G be a DAG and K be a PDAG. K is a completed PDAG (or CPDAG) with respect to the equivalence class of G : if (i) $\text{skeleton}(K) = \text{skeleton}(G)$; (ii) K contains a directed edge $X \rightarrow Y$ if and only if any DAG G' that is Markov equivalent to G contains the same directed edge $X \rightarrow Y$.

In other words, if an edge is directed in a CPDAG, all DAGs in the equivalent class agree on the direction of this edge. If an edge is undirected, then there are at least two DAGs within the equivalence class, such that they disagree on the direction of this edge.

Given a distribution P that is faithful to G , we can only identify the CPDAG of G . The population PC algorithm takes a distribution P as input and return a CPDAG. The algorithm starts from a complete, undirected graph and recursively deletes edges based on conditional independence decisions. This yields an undirected skeleton graph which can then be partially directed and further extended to represent the CPDAG. The algorithm has two parts: (i) identify the DAG skeleton; (ii) identify the CPDAG. After introducing the population PC algorithm, We describe a sample version PC algorithm that can reliably recover the true CPDAG purely based on observational data for Gaussian models.

18.9.2 PC Algorithm, Step 1: Identifying the Skeleton

Let P be a distribution that is faithful to G . We want to construct an undirected graph S such that $S = \text{skeleton}(G)$. The algorithm is based on evaluating conditional independence relationships of the form

$$X_i \perp\!\!\!\perp X_j \mid A \tag{18.79}$$

for different subsets of variables A . For finite data, these evaluations are based on statistical tests of conditional independence. The basic idea is if X_i and X_j are adjacent in G , then $X_i \perp\!\!\!\perp X_j \mid A$ does not hold for any A . This is summarized in the next theorem.

18.80 Theorem. Let G be a DAG and P be a distribution that is faithful to G . If X_i and X_j are adjacent in G , then the conditional independence test $X_i \perp\!\!\!\perp X_j \mid A$ fails for all $A \subset V \setminus \{i, j\}$. On the other hand, if X_i and X_j are not adjacent in G , then either $X_i \perp\!\!\!\perp X_j \mid \pi(X_i)$ or $X_i \perp\!\!\!\perp X_j \mid \pi(X_j)$, where $\pi(X_i), \pi(X_j)$ are the parent sets of X_i and X_j in the DAG G .

Proof. For the first part, with no loss of generality, we assume a directed edge $X \rightarrow Y$ is in

G . For any $A \subset V \setminus \{i, j\}$, there is no way for A to d-separate X_i and X_j . Since P is faithful to G , we know that X_i and X_j must be conditionally dependent for any $A \subset V \setminus \{i, j\}$.

For the second part, we consider two cases: (i) X_j is a descendant of X_i and (ii) X_j is not a descendant of X_i . By the definition of d-separation, in the first case we can show that $X_i \perp\!\!\!\perp X_j | \pi(X_j)$, and in the second case we have $X_i \perp\!\!\!\perp X_j | \pi(X_i)$. \square

From Theorem 18.80, we see that to determine whether X_i and X_j has an undirected edge in $\text{skeleton}(G)$, we need to check whether there exists $A \subset V \setminus \{i, j\}$, such that $X_i \perp\!\!\!\perp X_j | A$. If we can find such an A , it is enough to ensure that X_i and X_j are not adjacent. The corresponding A is called a *separating set* for X_i and X_j . In the next section, we will show that these separating sets are useful for determining unshielded colliders. If X_i and X_j are conditionally independent given A , they must be conditionally independent given any superset of A . Therefore, we can search the set of possible separating sets in order of increasing size.

To make sure that X_i and X_j are actually adjacent in G , we need to exhaustively check that $X_i \perp\!\!\!\perp X_j | A$ fails for all $A \subset V \setminus \{i, j\}$. This is computationally intensive. The corresponding algorithm is shown in Figure 18.17, where $\text{adj}(C, i)$ represents the adjacency nodes of X_i in an undirected graph C . The outer loop $k = 0, 1, \dots, d$ indexes the size of the separating sets. In the next paragraph, we will show that the number of iterations of this outer loop is upper bounded by the maximum degree of $\text{skeleton}(G)$. Also, it is easy to see that if $\text{skeleton}(G)$ is sparse, the algorithm will converge much faster.

Let $K \equiv$ the maximum degree of $\text{skeleton}(G)$. The following theorem shows the correctness of Algorithm 18.17.

18.81 Theorem. *Let G be a DAG and P be a distribution that is faithful to G . Then the algorithm in Figure 18.17 correctly reconstruct $\text{skeleton}(G)$. Furthermore, let*

$$\ell_{\text{stop}} \equiv \text{the maximum reached value of } \ell \text{ in the outer loop.} \quad (18.82)$$

Then either $\ell_{\text{stop}} = K - 1$ or $\ell_{\text{stop}} = K$.

Proof. Let C_{out} be the final output from the algorithm. We know the algorithm only removes an edge $X_i — X_j$ from the current skeleton graph if a separating set $A \subset V \setminus \{i, j\}$ is found such that $X_i \perp\!\!\!\perp X_j | A$. From Theorem 18.80, we know that all the removed edges do not belong to $\text{skeleton}(G)$. Therefore $\text{skeleton}(G) \subset C_{\text{out}}$. For the other direction, if for any edge $X_i — X_j \in C_{\text{out}}$, it follows that in G , X_i and X_j are not d-separated given any subset of the adjacencies of X_i or any adjacencies of X_j in C_{out} . Since the adjacencies of X_i is a superset of $\pi(X_i)$ and the adjacencies of X_j is a superset of $\pi(X_j)$, X_i and X_j are not d-separated given $\pi(X_i)$ and $\pi(X_j)$ in G . It then follows from Theorem 18.80 that X_i and X_j must be adjacent in G .

We now show that $\ell_{\text{stop}} = K - 1$ or $\ell_{\text{stop}} = K$. Since we just proved that $C_{\text{out}} = \text{skeleton}(G)$, it's obvious that $\ell_{\text{stop}} \leq K$. We now show that $\ell_{\text{stop}} \geq K - 1$. Suppose the contrary. Then $\ell_{\text{stop}} \leq K - 2$. We could then continue with a further iteration in the

Population PC Algorithm, Step 1: Skeleton Identification

Input: Vertex set V and joint distribution P .

Initialize $C \leftarrow$ complete undirected graph on V ; $\mathcal{A}_{ij} \leftarrow \Phi$ for any $i, j \in V$.

Loop through $k = 0, 1, \dots, d$:

- **Loop** through any two adjacent nodes X_i, X_j , such that $|\text{adj}(C, i) \setminus \{j\}| \geq k$:

- **For** every $A \in |\text{adj}(C, i) \setminus \{j\}|$ with $|A| = k$

If $X_i \perp\!\!\!\perp X_j \mid A$

* Remove $X_i - X_j$ from C ;

* $\mathcal{A}_{ij} \leftarrow A$;

* **Break**;

End if

- **End for**

Output: Estimated skeleton C and the class of separating sets \mathcal{A} .

Figure 18.17. The population PC algorithm on skeleton identification.

algorithm since $\ell_{stop} + 1 \leq K - 1$ and there is at least one node X_j with neighborhood-size $|\text{adj}(G, j)| = K$; that is, the reached stopping level would at least be $K - 1$ which contradicts the assumption that $\ell_{stop} \leq K - 2$. \square

To understand the time complexity of the PC algorithm for skeleton identification, the algorithm uses N conditional independence checks where N is at most

$$N \leq \binom{d}{2} \sum_{k=1}^K \binom{d-1}{k} \leq \frac{d^{K+1}}{2(d-1)}. \quad (18.83)$$

Here K is the maximal degree of any vertex in $\text{skeleton}(G)$; the worst case complexity is thus exponential.

18.9.3 PC Algorithm, Step 2: Identifying the CPDAG

Once the skeleton C_{out} and the class of separating sets \mathcal{A} are obtained, the second step of the PC algorithm is to identify all the unshielded colliders and further identify the CPDAG. We consider all of the triples $X_i - X_k - X_j$ with X_i and X_j nonadjacent in C_{out} as *candi-*

date unshielded colliders. The following theorem gives necessary and sufficient conditions under which a candidate is actually an unshielded collider.

18.84 Theorem. Let G be a DAG and P be a distribution that is faithful to G . Assume we use P as input and the PC algorithm part 1 outputs an identified skeleton C and a class of separating sets \mathcal{A} . A candidate unshielded collider $X_i—X_k—X_j$ in C is an unshielded collider $X_i \rightarrow X_k \leftarrow X_j$ if and only if $X_k \notin \mathcal{A}_{ij}$.

Proof. Since X_i and X_j are nonadjacent, there exists a non-empty set $A = \mathcal{A}_{ij}$ such that $X_i \perp\!\!\!\perp X_j | A$. First, we show that if $X_i—X_k—X_j$ is an unshielded collider $X_i \rightarrow X_k \leftarrow X_j$, then $X_k \notin A$. Suppose on the contrary that $X_k \in A$, then conditioning on A , since X_k is also conditioned on, there is no way for X_i and X_j to be d-separated. Thus it is impossible that $X_i \perp\!\!\!\perp X_j | A$, which leads to contradiction.

Next, we show that if $X_i—X_k—X_j$ is not an unshielded collider, then $X_k \in A$. To see this, since $X_i—X_k—X_j$ is not an unshielded collider, there are only three possible cases: $X_i \rightarrow X_k \rightarrow X_j$, $X_i \leftarrow X_k \leftarrow X_j$, and $X_i \leftarrow X_k \rightarrow X_j$. In any case, to d-separate X_i and X_j , X_k must be conditioned on. Therefore, we have $X_k \in A$. \square

From Theorem 18.84, it is easy for us to evaluate a candidate unshielded collider $X_i—X_k—X_j$; we only need to check if X_k belongs to the separating set of X_i and X_j .

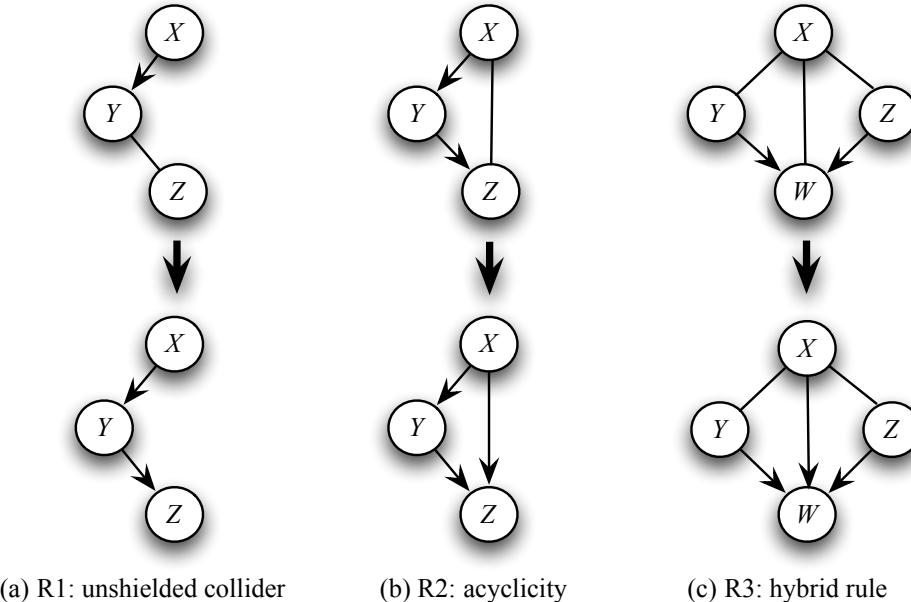


Figure 18.18. Three basic configurations for which directions of certain edges can be determined: (a) R1: unshielded collider rule; (b) R2: acyclicity rule. (c) R3: hybrid rule.

Once the skeleton and all the unshielded colliders are given, the next task is to identify

the whole CPDAG. This requires us to infer the directions of certain edges based on existing knowledge. Three simple rules are provided in Figure 18.18. The first rule R1 is based on the idea that if a local configuration $X \rightarrow Y - Z$ is not an unshielded collider, then the edge connecting Y and Z must be $Y \rightarrow Z$. Otherwise, we will form an unshielded collider. The second rule R2 is based on the idea that the graph is a DAG, thus no cycles are allowed. The third rule R3 is in fact a combined application of R1 and R2. To see this, given the upper configuration in R3, let's assume we have a directed edge $W \rightarrow X$. Then by R2, we know that the directions of edge $Y \rightarrow X$ and $Z \rightarrow X$ must be $Y \rightarrow X$ and $Z \rightarrow X$. This forms an unshielded collider $Y \rightarrow X \leftarrow Z$, which violates R1. These three rules can be applied in a dynamic way. Given a PDAG, a rule applies whenever a subgraph in the PDAG can be found that matches the upper parts of the three rules. In that case, we modify this subgraph by applying the rule and orienting a previously undirected edge. Such an operation then creates a new PDAG. The new PDAG may create another subset that match one of the three rules, leading to the orientation of more edges. We could then proceed this process until we obtain a PDAG on which none of these three rules can apply. This PDAG is then exported as the final identified CPDAG. The convergence of this procedure is obvious. We will show its correctness in a later theorem.

Population PC Algorithm, Step 2: CPDAG Identification

Input: The identified skeleton C and a class of separating sets \mathcal{A} .

Initialize $K \leftarrow C$.

For every pair of nonadjacent variables X_i, X_j with common neighbor X_k :

If $k \notin \mathcal{A}_{ij}$ **Then** Replace $X_i - X_k - X_j$ by $X_i \rightarrow X_k \leftarrow X_j$;

End for

Loop until converge:

Find a subgraph in K on which any rule R1-R3 in Figure 18.18 can apply;

Apply the rule on the subgraph and add in the corresponding directions;

End loop

Output: Identified CPDAG K .

Figure 18.19. The population PC algorithm on CPDAG identification.

The algorithm in Figure 18.19 summarizes the whole process. With the estimated skeleton and the class of separating sets as input. The algorithm have two parts. In the first part, every candidate unshielded colliders are examined and all unshielded colliders are identi-

fied. In the section part, many undirected edges in the obtained PDAG are further oriented until no updates can be made. The final PDAG is then output as the identified CPDAG.

The next theorem secures the correctness of the algorithm in Figure 18.19, which was first proved by Meek (1995).

18.85 Theorem. *Let G be a DAG and P be a distribution that is faithful to G . Then the algorithm in Figure 18.19 correctly reconstruct the CPDAG of G .*

Proof Idea. In Theorem 18.81, we have already shown that the identified skeleton is correct. We only need to show all the directed and undirected edges in the output CPDAG K is indeed the same as the true CPDAG of G . To establish this result we need to show three properties of the obtained PDAG: (i) Property 1: the final graph returned by the algorithm is acyclic; (ii) If an edge $X \rightarrow Y$ appears in K , then this edge appears in all DAGs of the equivalence class of G ; (iii) If an undirected edge $X-Y \in K$, then we can find two DAGs G_1 and G_2 . Both G_1 and G_2 are Markov equivalent to G and $X \rightarrow Y \in G_1$ and $X \leftarrow Y \in G_1$. The first two properties are straightforward. The third property requires additional machinery and is omitted here. More details can be found in (Meek, 1995). \square

The time complexity of the second part of the PC algorithm is no larger than that of the first part on skeleton identification. Therefore, the total time complexity of the algorithm is $O(d^{K+1})$. However, one thing to keep in mind is that in applications, such a worst case scenario is seldom met.

18.9.4 PC Algorithm: Sample Version

In the population version PC algorithm, the only place where we utilize the population quantities is on the conditional independence query $X_i \perp\!\!\!\perp X_j | A$ with $A \subset V \setminus \{i, j\}$. If we only have observed data points, we need to test whether X_i and X_j are conditional independent given A . For Gaussian DAGs, such a test is very easy to construct.

Under the Gaussian DAG assumption, Theorem 18.29 says that

$$X_i \perp\!\!\!\perp X_j | A \text{ if and only if } \rho_{i,j|A} = 0, \quad (18.86)$$

where $\rho_{i,j|A}$ is the partial correlation between X_i and X_j given A . Therefore, to query whether $X_i \perp\!\!\!\perp X_j | A$, we only need to test the hypothesis

$$H_0 : \rho_{i,j|A} = 0 \text{ vs. } H_1 : \rho_{i,j|A} \neq 0. \quad (18.87)$$

Let $\widehat{\rho}_{i,j|A}$ be the sample partial correlation which can be calculated using the recursive formula (18.31). In order to test whether $\widehat{\rho}_{i,j|A} = 0$, we apply Fisher's Z -transform and define

$$\widehat{z}_{ij|A} = \frac{1}{2} \log \left(\frac{1 + \widehat{\rho}_{i,j|A}}{1 - \widehat{\rho}_{i,j|A}} \right) \text{ and } z_{ij|A} = \frac{1}{2} \log \left(\frac{1 + \rho_{i,j|A}}{1 - \rho_{i,j|A}} \right). \quad (18.88)$$

Classical distribution theory in the Gaussian case characterizes the asymptotic distribution of $\widehat{z}_{ij|A}$:

$$(\sqrt{n - |A| - 3})(\widehat{z}_{ij|A} - z_{ij|A}) \xrightarrow{D} N(0, 1), \quad (18.89)$$

where $|A|$ is the cardinality of A . Under the null hypothesis we have $z_{ij|A} = 0$, which suggests a level α test that we reject the null hypothesis if

$$(\sqrt{n - |A| - 3})|\widehat{z}_{ij|A}| > \Phi^{-1}\left(1 - \frac{\alpha}{2}\right), \quad (18.90)$$

where $\Phi(\cdot)$ is the cumulative function for the standard Gaussian random variable. The sample PC algorithm is almost identical to the population PC algorithm with the population conditional independence query on whether $X_i \perp\!\!\!\perp X_j | A$ replaced by a finite sample level α test (18.90).

Sample Version of the PC Algorithm

The Sample version PC algorithm is identical to the population PC algorithm but replace the conditional independence query on whether $X_i \perp\!\!\!\perp X_j | A$ by a finite sample level α test:

$$(\sqrt{n - |A| - 3})|\widehat{z}_{ij|A}| > \Phi^{-1}\left(1 - \frac{\alpha}{2}\right). \quad (18.91)$$

Figure 18.20. The sample version PC algorithm.

The large-sample properties of this sample version PC algorithm has been analyzed by Kalisch and Bühlmann (2007). They have the following assumptions:

- (A1) The distribution P is multivariate Gaussian and faithful to the DAG G even when the dimension d increases with the sample n .
- (A2) $d = O(n^a)$ for some $0 \leq a < \infty$.
- (A3) Let $q = \max_{1 \leq j \leq d} |\text{adj}(G, j)|$. Then $q = O(n^{1-b})$ for some $0 < b \leq 1$.
- (A4) $\inf\{|\rho_{i,j|A}|; i, j, A \text{ with } \rho_{i,j|A} \neq 0\} \geq c_n$, where $c_n^{-1} = O(n^\gamma)$ for some $0 < \gamma < b/2$. Also $\sup_{i,j,A} |\rho_{i,j|A}| \leq M < 1$. Here $0 < b \leq 1$ is as in (A3).

18.92 Theorem. (Kalisch and Bühlmann (2007)) *Under Assumptions (A1) - (A4), we denote by Γ the true CPDAG. Let $\widehat{\Gamma}_n^{\alpha_n}$ be the estimated CPDAG from the sample PC algorithm*

with sample size n and level α_n for each conditional independence test. Then, there exists $\alpha_n \rightarrow 0$ as n goes to infinity, such that

$$\mathbb{P}(\widehat{\Gamma}_n^{\alpha_n} = \Gamma) = 1 - O(\exp(-Cn^{1-2d})) \rightarrow 1 \text{ as } n \rightarrow \infty \text{ for some } 0 < C < \infty,$$

where $d > 0$ is as in (A4).

18.9.5 Analysis of Cell-Signaling Networks

We apply the PC algorithm on a flow-cytometry dataset from Sachs et al. (2005) with $p = 11$ variables and $n = 7,466$ data points. Each data point corresponds to a cell and the variables correspond to the expression levels of proteins. The abbreviated variable names are: Raf, Mek, Plcg, PIP2, PIP3, P44/42, Akt, PKA, PKC, P38, Jnk. The only tuning parameter for the PC algorithm is the level α of the conditional independence tests. The larger the value α is, the sparser the estimated CPDAG will be. In this example, we use $\alpha = 0.01$ and the estimated CPDAG is shown in Figure 18.21.

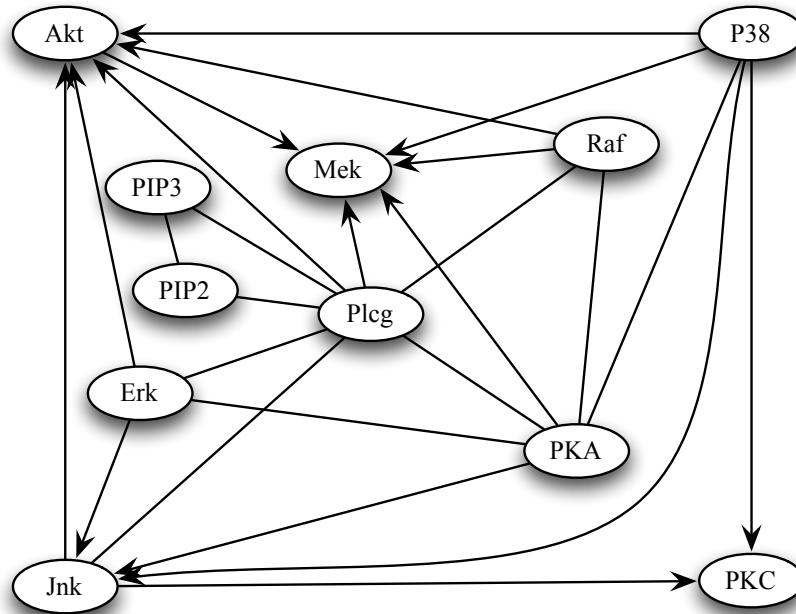


Figure 18.21. An estimated CPDAG from a flow cytometry dataset, with $d = 11$ protein measured on $n = 7,466$ cells. The network structure is estimated using the PC algorithm with level $\alpha = 0.01$.

In this example, the estimated CPDAG contains many undirected edges. We see that the variable Mek is pointed to by all variables that are connected with it. In the next chapter, we will estimate undirected Gaussian graphs on this same dataset, and discuss the relationships between directed acyclic graphs and undirected graphs.

18.10 Bibliographic Remarks

There are a number of texts on DAGs including Edwards (1995) and Jordan (2003). The first use of DAGs for representing causal relationships was by Wright (1934). Modern treatments are contained in Spirtes et al. (2000) and Pearl (2000). Robins et al. (2003) discuss the problems with estimating causal structure from data. Nice treatments on graphical model inference appears in Bishop (2007) and Alpaydin (2004). A very thorough and excellent discussion of DAGs can be found in Koller and Friedman (2009).

Exercises

18.1 Complete the proof of Theorem 18.5.

18.2 Show the equivalence of the following two statements:

- $p(x | y, z) = p(x | z)$ for all x, y and z
- $p(x, y | z) = p(x | z)p(y | z)$ for all x, y and z .

18.3 Prove Theorem 18.17.

18.4 Prove Theorem 18.26.

18.5 Prove Proposition 18.30.

18.6 Let X, Y and Z have the following joint distribution:

	$Y = 0$	$Y = 1$		$Y = 0$	$Y = 1$
$X = 0$.405	.045	$X = 0$.125	.125
$X = 1$.045	.005	$X = 1$.125	.125
	$Z = 0$			$Z = 1$	

(a) Find the conditional distribution of X and Y given $Z = 0$ and the conditional distribution of X and Y given $Z = 1$.

(b) Show that $X \perp\!\!\!\perp Y | Z$.

(c) Find the marginal distribution of X and Y .

(d) Show that X and Y are not marginally independent.

18.7 Consider the three DAGs in Figure 18.12 without a collider. Prove that $X \perp\!\!\!\perp Z | Y$.

18.8 Consider the DAG in Figure 18.12 with a collider. Prove that $X \perp\!\!\!\perp Z$ and that X and Z are dependent given Y .

18.9 Let $X \in \{0, 1\}$, $Y \in \{0, 1\}$, $Z \in \{0, 1, 2\}$. Suppose the distribution of (X, Y, Z) is Markov to: $X \rightarrow Y \rightarrow Z$. Create a joint distribution $p(x, y, z)$ that is Markov to this DAG. Generate 1000 random vectors from this distribution. Estimate the distribution from the data using maximum likelihood. Compare the estimated distribution to the true distribution. Let $\theta = (\theta_{000}, \theta_{001}, \dots, \theta_{112})$ where $\theta_{rst} = \mathbb{P}(X = r, Y = s, Z = t)$. Use the bootstrap to get standard errors and 95 percent confidence intervals for these 12 parameters.

Causal Inference

Prediction and causation are very different. Typical questions are:

Prediction: Predict Y after **observing** $X = x$

Causation: Predict Y after **setting** $X = x$.

Causation involves predicting the effect of an intervention. For example:

Prediction: Predict health given that a person takes vitamin C

Causation: Predict health if I give a person vitamin C

The difference between passively observing $X = x$ and actively intervening and setting $X = x$ is significant and requires different techniques and, typically, much stronger assumptions. This is the area known as *causal inference*.

For years, causal inference was studied by statisticians, epidemiologists and economists. The machine learning community was largely uninterested. This has changed. The ML community now has an active research program in causation. This is because it is now recognized that many problems that were once treated as prediction problems are actually causal questions. Questions like: “If I place this ad on a web page, will people click on it?” and “If I recommend a product will people buy it?” are causal questions, not predictive questions.

1 Preliminaries

Before we jump into the details, there are a few general concepts to discuss.

1.1 Two Types of Causal Questions

There are two types of causal questions. The first deals with questions like this: do cell phones cause brain cancer? In this case, there are variables X and Y and we want to know the causal effect of X on Y . The challenges are: find a parameter θ that characterizes the causal influence of X on Y and find a way to estimate θ . This is usually what we mean when we refer to *causal inference*.

The second question is: given a set of variables, determine the causal relationship between the variables. This is called *causal discovery*. **As we shall see, this problem is statistically impossible** despite the large number of papers on the topic.

1.2 Two Types of Data

Data can be from a controlled, randomized experiment or from an observational study. In the former, X is randomly assigned to subjects. In the latter, it is not randomly assigned. In randomized experiments, causal inference is straightforward. In observational (non-randomized) studies, the problem is much harder and requires stronger assumptions and also requires subject matter knowledge. Statistics and Machine Learning cannot solve causal problems without background knowledge.

1.3 Two Languages for Causation

There are two different mathematical languages for studying causation. The first is based on *counterfactuals*. The second is based on *causal graphs*. It will not seem obvious at first, but the two are mathematically equivalent (apart from some small details). Actually, there is a third language called *structural equation models* but this is very closely related to causal graphs.

1.4 Example

Consider this story. A mother notices that tall kids have a higher reading level than short kids. The mother puts her small child on a device and stretches the child until he is tall. She is dismayed to find out that his reading level has not changed.

The mother is correct that height and reading skill are **associated**. Put another way, you can use height to predict reading skill. But that does not imply that height *causes* reading skill. This is what statisticians mean when they say:

correlation is not causation.

On the other hand, consider smoking and lung cancer. We know that smoking and lung cancer are associated. But we also believe that smoking causes lung cancer. In this case, we recognize that intervening and forcing someone to smoke does change his probability of getting lung cancer.

1.5 Prediction Versus Causation

The difference between prediction (association/correlation) and causation is this: in prediction we are interested in

$$\mathbb{P}(Y \in A | X = x)$$

which means: the probability that $Y \in A$ given that we **observe** that X is equal to x . For causation we are interested in

$$\mathbb{P}(Y \in A | \text{set } X = x)$$

which means: the probability that $Y \in A$ given that we **set** X equal to x . Prediction is about passive observation. Causation is about active intervention. The phrase **correlation**

is not causation can be written mathematically as

$$\mathbb{P}(Y \in A|X = x) \neq \mathbb{P}(Y \in A|\text{set } X = x).$$

Despite the fact that causation and association are different, people confuse them up all the time, even people trained in statistics and machine learning. On TV recently there was a report that good health is associated with getting seven hours of sleep. So far so good. Then the reporter goes on to say that, therefore, everyone should strive to sleep exactly seven hours so they will be healthy. Wrong. That's confusing causation and association. Another TV report pointed out a correlation between people who brush their teeth regularly and low rates of heart disease. An interesting correlation. Then the reporter (a doctor in this case) went on to urge people to brush their teeth to save their hearts. Wrong!

To avoid this confusion we need a way to discuss causation mathematically. That is, we need somehow to make $\mathbb{P}(Y \in A|\text{set } X = x)$ formal. As I mentioned earlier, there are two common ways to do this. One is to use **counterfactuals**. The other is to use **causal graphs**. There are two different languages for saying the same thing.

Causal inference is tricky and should be used with great caution. The main messages are:

1. Causal effects can be estimated consistently from randomized experiments.
2. It is difficult to estimate causal effects from observational (non-randomized) experiments.
3. All causal conclusions from observational studies should be regarded as very tentative.

Causal inference is a vast topic. We will only touch on the main ideas here.

2 Counterfactuals

Consider two variables X and Y . We will call X the “exposure” or the “treatment.” We call Y the “response” or the “outcome.” For a given subject we see (X_i, Y_i) . What we don’t see is what their value of Y_i would have been if we changed their value of X_i . This is called the counterfactual. The whole causal story is made clear in Figure 1 which shows data (left) and the counterfactuals (right).

Suppose that X is a binary variable that represents some exposure. So $X = 1$ means the subject was exposed and $X = 0$ means the subject was not exposed. We can address the problem of predicting Y from X by estimating $\mathbb{E}(Y|X = x)$. To address causal questions, we introduce *counterfactuals*. Let Y_1 denote the response if the subject is exposed. Let Y_0 denote the response if the subject is not exposed. Then

$$Y = \begin{cases} Y_1 & \text{if } X = 1 \\ Y_0 & \text{if } X = 0. \end{cases}$$

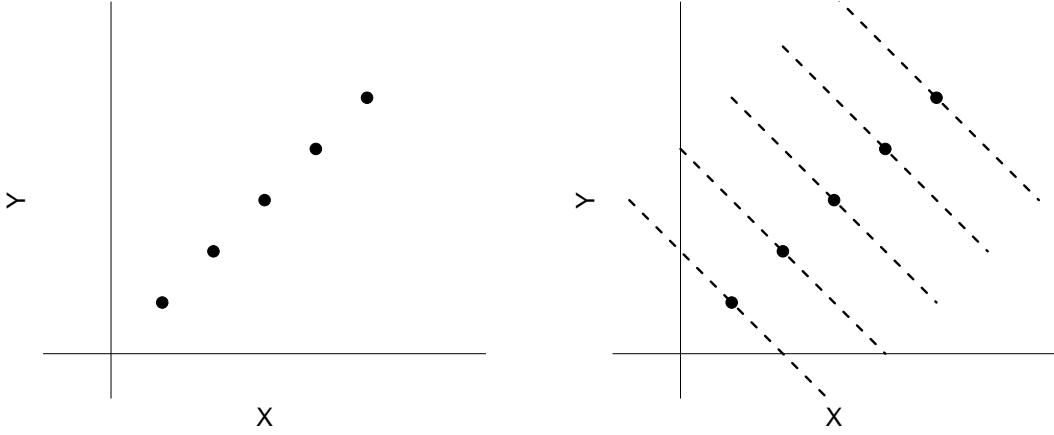


Figure 1: *Left:* X and Y have positive association. *Right:* The lines are the counterfactuals, i.e. what would happen to each person if I changed their X value. Despite the positive association, the causal effect is negative. If we increase X everyone's Y values will decrease.

More succinctly

$$Y = XY_1 + (1 - X)Y_0. \quad (1)$$

We have replaced the random variables (X, Y) with the more detailed variables (X, Y_0, Y_1, Y) where $Y = XY_1 + (1 - X)Y_0$. When X is continuous, the counterfactual is a function $Y(\cdot)$. Then $Y(x)$ is value of the function $Y(\cdot)$ when $X = x$. The observed Y is $Y \equiv Y(X)$.

If we expose a subject, we observe Y_1 but we do not observe Y_0 . Indeed, Y_0 is the value we would have observed if the subject had been exposed. The unobserved variable is called a *counterfactual*. The variables (Y_0, Y_1) are also called *potential outcomes*. We have enlarged our set of variables from (X, Y) to (X, Y, Y_0, Y_1) . A small dataset might look like this:

X	Y	Y_0	Y_1
1	1	*	1
1	1	*	1
1	0	*	0
1	1	*	1
0	1	1	*
0	0	0	*
0	1	1	*
0	1	1	*

The asterisks indicate unobserved variables. Causal questions involve the distribution $p(y_0, y_1)$ of the potential outcomes. We can interpret $p(y_1)$ as $p(y|\text{set } X = 1)$ and we can

interpret $p(y_0)$ as $p(y|\text{set } X = 0)$. The *mean treatment effect* or *mean causal effect* is defined by

$$\theta = \mathbb{E}(Y_1) - \mathbb{E}(Y_0) = \mathbb{E}(Y|\text{set } X = 1) - \mathbb{E}(Y|\text{set } X = 0).$$

The parameter θ has the following interpretation: θ is the mean response if we exposed everyone minus the mean response if we exposed no-one.

Lemma 1 *In general,*

$$\mathbb{E}[Y_1] \neq \mathbb{E}[Y|X = 1] \quad \text{and} \quad \mathbb{E}[Y_0] \neq \mathbb{E}[Y|X = 0].$$

Exercise: Prove this.

Suppose now that we observe a sample $(X_1, Y_1), \dots, (X_n, Y_n)$. Can we estimate θ ? In general the answer is no. We can estimate

$$\alpha = \mathbb{E}(Y|X = 1) - \mathbb{E}(Y|X = 0)$$

but α is not equal to θ . Quantities like $\mathbb{E}(Y|X = 1)$ and $\mathbb{E}(Y|X = 0)$ are predictive parameters. These are things that are commonly estimated in statistics and machine learning.

Let's formalize this. Let \mathcal{P} be the set of distributions for (X, Y_0, Y_1, Y) such that $P(X = 0) > \delta$ and $P(X = 1) > \delta$ for some $\delta > 0$. (We have no hope if we do not have positive probability of observing exposed and unexposed subjects.) Recall that $Y = XY_1 + (1-X)Y_0$. The observed data are $(X_1, Y_1), \dots, (X_n, Y_n) \sim P$. Let $\theta(P) = \mathbb{E}[Y_1] - \mathbb{E}[Y_0]$. An estimator is uniformly consistent if, for every $\epsilon > 0$,

$$\sup_{P \in \mathcal{P}} P(|\hat{\theta}_n - \theta(P)| > \epsilon) \rightarrow 0$$

as $n \rightarrow \infty$.

Theorem 2 *In general, there does not exist a uniformly consistent estimator of θ .*

Proof. It is easy construct $p(x, y_0, y_1)$ and $q(x, y_0, y_1)$ such that $\theta(p) \neq \theta(q)$ and yet $p(x, y) = q(x, y)$. ■

In the case that X is continuous, the causal quantity (or rather, an example of a causal quantity) is

$$\theta(x) = \mathbb{E}[Y(x)]$$

which, in general, is NOT equal to $m(x) = \mathbb{E}[Y|X = x]$.

2.1 Two Ways to Make θ Estimable

Fortunately, there are two ways¹ to make θ estimable. The first is randomization and the second is adjusting for confounding.

Randomization. Suppose that we randomly assign X . Then X will be independent of (Y_0, Y_1) . In symbols:

$$\text{random treatment assignment implies : } (Y_0, Y_1) \perp\!\!\!\perp X.$$

Of course, we can't estimate θ if we always assign $X = 1$ or $X = 0$. We assume that $0 < \delta \leq P(X = 1) \leq 1 - \delta < 1$ for some δ . Let \mathcal{P} be all such distributions.

Warning! Note that X is not independent of Y .

Theorem 3 *If X is randomly assigned, then $\theta = \alpha$ where*

$$\alpha = \mathbb{E}(Y|X = 1) - \mathbb{E}(Y|X = 0).$$

A uniformly consistent estimator of α (and hence θ) is the plug-in estimator

$$\hat{\alpha} = \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i} - \frac{\sum_{i=1}^n (1 - X_i) Y_i}{\sum_{i=1}^n (1 - X_i)}.$$

That is, for every $\epsilon > 0$,

$$\sup_{P \in \mathcal{P}} P(|\hat{\alpha} - \theta| > \epsilon) \rightarrow 0$$

as $n \rightarrow \infty$.

Proof. Since X is independent of (Y_0, Y_1) , we have

$$\begin{aligned} \alpha &= \mathbb{E}(Y|X = 1) - \mathbb{E}(Y|X = 0) \\ &= \mathbb{E}(Y_1|X = 1) - \mathbb{E}(Y_0|X = 0) \quad \text{since } Y = XY_1 + (1 - X)Y_0 \\ &= \mathbb{E}(Y_1) - \mathbb{E}(Y_0) = \theta \quad \text{since } (Y_0, Y_1) \perp\!\!\!\perp X. \end{aligned}$$

Hence, random assignment makes θ equal to α . To prove the consistency of $\hat{\alpha}$, note that we can write $\hat{\alpha} = (A_n/B_n) - (C_n/D_n)$. Also note that

$$\alpha = \frac{\mathbb{E}[YX]}{\mathbb{E}[X]} - \frac{\mathbb{E}[Y(1 - X)]}{\mathbb{E}[1 - X]} \equiv \frac{A}{B} - \frac{C}{D}.$$

Let ϵ be a small positive constant. By Hoeffding's inequality and the union bound, with high probability, $A_n/B_n < (A + \epsilon)/(B - \epsilon) < (A/B) + \epsilon\Delta_1$ for some positive constant Δ_1 .

¹A third way is to use instrumental variables but we won't discuss that.

Similarly, $A_n/B_n > (A/B) - \epsilon\Delta_2$, say. A similar argument applies to the second term and the result follows. ■

Similarly, we can construct a test ϕ for testing $H_0 : \theta = 0$ versus $H_1 : \theta \neq 0$ such that we have type I error control

$$\sup_{P \in \mathcal{P}_0} P(\phi = 1) \leq \alpha$$

and with non-trivial power: for any $\epsilon > 0$,

$$\inf_{P \in \mathcal{P}_\epsilon} P(\phi = 1) \rightarrow 1$$

where \mathcal{P}_ϵ is the set of distribution with $|\theta| \geq \epsilon$. We can also construct a confidence set (using Hoeffding's inequality or the CLT) such that

$$\inf_{P \in \mathcal{P}} P(\theta \in C) \geq 1 - \alpha.$$

To summarize: **If X is randomly assigned then correlation = causation.** This is why people spend millions of dollars doing randomized experiments.

The same results hold when X is continuous. In this case there is a counterfactual $Y(x)$ for each value x of X . We again have that, in general,

$$\mathbb{E}[Y(x)] \neq \mathbb{E}[Y|X = x].$$

See Figure 1. But if X is randomly assigned, then we do have $\mathbb{E}[Y(x)] = \mathbb{E}[Y|X = x]$ and so $\mathbb{E}[Y(x)]$ can be consistently estimated using standard regression methods. Indeed, if we had randomly chosen the X values in Figure 1 then the plot on the left would have been downward sloping. To see this, note that $\theta(x) = \mathbb{E}[Y(x)]$ is defined to be the average of the lines in the right plot. Under randomization, X is independent of $Y(x)$. So

$$\text{right plot} = \theta(x) = \mathbb{E}[Y(x)] = \mathbb{E}[Y(x)|X = x] = \mathbb{E}[Y|X = x] = \text{left plot}.$$

In other words, under randomization, $\theta(x) = m(x)$ where $m(x) = \mathbb{E}(Y|X = x)$ is the usual regression function. So you can use everything you know about regression estimation and then you are estimating the causal effect.

Adjusting For Confounding. In some cases it is not feasible to do a randomized experiment and we must use data from observational (non-randomized) studies. Smoking and lung cancer is an example. Can we estimate causal parameters from observational (non-randomized) studies? The answer is: sort of.

In an observational study, the treated and untreated groups will not be comparable. Maybe the healthy people chose to take the treatment and the unhealthy people didn't. In

other words, X is not independent of (Y_0, Y_1) . The treatment may have no effect but we would still see a strong association between Y and X . In other words, α might be large even though $\theta = 0$.

Here is a simplified example. Suppose X denotes whether someone takes vitamins and Y is some binary health outcome (with $Y = 1$ meaning “healthy.”)

X	1	1	1	1	0	0	0	0
Y_0	1	1	1	1	0	0	0	0
Y_1	1	1	1	1	0	0	0	0

In this example, there are only two types of people: healthy and unhealthy. The healthy people have $(Y_0, Y_1) = (1, 1)$. These people are healthy whether or not they take vitamins. The unhealthy people have $(Y_0, Y_1) = (0, 0)$. These people are unhealthy whether or not they take vitamins. The observed data are:

X	1	1	1	1	0	0	0	0
Y	1	1	1	1	0	0	0	0

In this example, $\theta = 0$ but $\alpha = 1$. The problem is that people who choose to take vitamins are different than people who choose not to take vitamins. That’s just another way of saying that X is not independent of (Y_0, Y_1) .

To account for the differences in the groups, we can measure **confounding variables**. These are the variables that affect both X and Y . These variables explain why the two groups of people are different. In other words, these variables account for the dependence between X and (Y_0, Y_1) . By definition, there are no such variables in a randomized experiment. The hope is that if we measure enough confounding variables $Z = (Z_1, \dots, Z_k)$, then, perhaps the treated and untreated groups will be comparable, conditional on Z . This means that X is independent of (Y_0, Y_1) conditional on Z . We say that there is *no unmeasured confounding*, or that *ignorability holds*, if

$$X \perp\!\!\!\perp (Y_0, Y_1) \mid Z.$$

The only way to measure the important confounding variables is to use subject matter knowledge. In other words, **causal inference in observational studies is not possible without subject matter knowledge**.

Theorem 4 Suppose that

$$X \perp\!\!\!\perp (Y_0, Y_1) \mid Z.$$

Then

$$\theta \equiv \mathbb{E}(Y_1) - \mathbb{E}(Y_0) = \int \mu(1, z)p(z)dz - \int \mu(0, z)p(z)dz \quad (2)$$

where

$$\mu(x, z) = \mathbb{E}(Y|X = x, Z = z).$$

A consistent estimator of θ is

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n \hat{\mu}(1, Z_i) - \frac{1}{n} \sum_{i=1}^n \hat{\mu}(0, Z_i)$$

where $\hat{\mu}(x, z)$ is an appropriate, consistent estimator of the regression function $\mu(x, z) = \mathbb{E}[Y|X = x, Z = z]$.

Remark: Estimating the quantity in (2) well is difficult and involves an area of statistics called *semiparametric inference*. In statistics, biostatistics, econometrics and epidemiology, this is the focus of much research. It appears that the machine learning community has ignored this goal and has focused instead on the quixotic goal of causal discovery.

Proof. We have

$$\begin{aligned} \theta &= \mathbb{E}(Y_1) - \mathbb{E}(Y_0) \\ &= \int \mathbb{E}(Y_1|Z = z)p(z)dz - \int \mathbb{E}(Y_0|Z = z)p(z)dz \\ &= \int \mathbb{E}(Y_1|X = 1, Z = z)p(z)dz - \int \mathbb{E}(Y_0|X = 0, Z = z)p(z)dz \\ &= \int \mathbb{E}(Y|X = 1, Z = z)p(z)dz - \int \mathbb{E}(Y|X = 0, Z = z)p(z)dz \end{aligned} \quad (3)$$

where we used the fact that X is independent of (Y_0, Y_1) conditional on Z in the third line and the fact that $Y = (1 - X)Y_1 + XY_0$ in the fourth line. ■

The process of including confounding variables and using equation (2) is known as *adjusting for confounders* and $\hat{\theta}$ is called the *adjusted treatment effect*. The choice of the estimator $\hat{\mu}(x, z)$ is delicate. If we use a nonparametric method then we have to choose the smoothing parameter carefully. Unlike prediction, bias and variance are not equally important. **The usual bias-variance tradeoff does not apply.** In fact bias is worse than variance and we need to choose the smoothing parameter smaller than usual. As mentioned above, there is a branch of statistics called *semiparametric inference* that deals with this problem in detail.

It is instructive to compare the causal effect

$$\theta = \int \mu(1, z)p(z)dz - \int \mu(0, z)p(z)dz$$

with the predictive quantity

$$\begin{aligned}\alpha &= \mathbb{E}(Y|X = 1) - \mathbb{E}(Y|X = 0) \\ &= \int \mu(1, z)p(z|X = 1)dz - \int \mu(0, z)p(z|X = 0)dz\end{aligned}$$

which are mathematically (and conceptually) quite different.

We need to treat $\hat{\theta}$ cautiously. It is very unlikely that we have successfully measured all the relevant confounding variables so $\hat{\theta}$ should be regarded as a crude approximation to θ at best.

In the case where $\mathbb{E}[Y|X = x, Z = z]$ is linear, the adjusted treatment effect takes a simple form. Suppose that $\mathbb{E}[Y|X = x, Z = z] = \beta_0 + \beta_1 x + \beta_2^T z$. Then

$$\theta = \int [\beta_0 + \beta_1 + \beta_2^T z]dP(z) - \int [\beta_0 + \beta_2^T z]dP(z) = \beta_1.$$

In a linear regression, the coefficient in front of x is the causal effect of x if (i) the model is correct and (ii) all confounding variables are included in the regression.

More generally,

$$\begin{aligned}\theta(x) &= \mathbb{E}[Y(x)] = \mathbb{E}[Y(x)|Z = z]dP(z) = \int \mathbb{E}[Y(x)|Z = z, X = x]dP(z) \\ &= \int \mathbb{E}[Y|Z = z, X = x]dP(z) = \int m(x, z)dP(z)\end{aligned}$$

where $m(x, z) = \mathbb{E}[Y|Z = z, X = x]$ is the usual regression function. We can insert an estimate \hat{m} and replace the integral over z with an average:

$$\hat{\theta}(x) = \frac{1}{n} \sum_i \hat{m}(x, Z_i).$$

However, you should not use cross-validation to choose the smoothing parameter. You need to use methods known as *semi-parametric inference* to get an accurate estimate.

An alternative is to use *matching* which I will explain in class.

3 Causal Graphs and Structural Equations

Another way to capture the difference between $P(Y \in A|X = x)$ and $P(Y \in A|\text{set } X = x)$ is to represent the distribution using a directed graph. Then we capture the second statement by performing certain operations on the graph. Specifically, we break the arrows into the some variables to represent an intervention.

A Directed Acyclic Graph (DAG) is a graph for a set of variables with no cycles. The graph defines a set of distributions of the form

$$p(y_1, \dots, y_k) = \prod p(y_j | \text{parents}(y_j))$$

where $\text{parents}(y_j)$ are the parents of y_j . A **causal graph** is a DAG with extra information. A DAG is a causal graph if it correctly encodes the effect of setting a variable to a fixed value.

Consider the graph G in Figure 2. Here, X denotes treatment, Y is response and Z is a confounding variable. To find the causal distribution $p(y|\text{set } X = x)$ we do the following steps:

1. Form a new graph G_* by removing all arrow into X . Now set X equal to x . This corresponds to replacing the joint distribution $p(x, y, z) = p(z)p(x|z)p(y|x, z)$ with the new distribution $p_*(y, z) = p(z)p(y|x, z)$. The factor $p(x|z)$ is removed because we know regard x as a fixed number. (Actually, $p(x|z)$ is replaced with a point mass at x .)
2. Compute the distribution of y from the new distribution:

$$p(y|\text{set } X = x) \equiv p_*(y) = \int p_*(y, z) dz = \int p(z)p(y|x, z) dz.$$

Now we have that

$$p(y|\text{set } X = 1) - p(y|\text{set } X = 0) = \int p(y|1, z) p(z) dz - \int p(y|0, z) p(z) dz.$$

Hence,

$$\begin{aligned} \theta &= \mathbb{E}[Y|\text{set } X = 1] - \mathbb{E}[Y|\text{set } X = 0] \\ &= \int y p(y|1, z) p(z) dz - \int y p(y|0, z) p(z) dz = \mathbb{E}[Y|X = 1, Z = z] p(z) dz - \mathbb{E}[Y|X = 0, Z = z] p(z) dz \\ &= \int \mu(1, z) p(z) dz - \int \mu(0, z) p(z) dz \end{aligned}$$

This is precisely the same equation as (2). Both approaches lead to the same formulas for the causal effect. Of course, if there were unobserved confounding variables, then the formula for θ would involve these variables and the causal effect would be non-estimable (as before).

In a randomized experiment, there would be no arrow from Z to X . (That's the point of randomization). In that case the above calculations shows that $\theta = \mathbb{E}(Y|X = 1) - \mathbb{E}(Y|X = 0)$ which again agrees with the counterfactual approach.

In general, the DAG approach and the counterfactual approach lead to the same formulas for causal effects. They are two different languages for the same thing.

The formulas derived from a causal graph will only be correct if the causal graph is correct. Right now, we are assuming that the the correct causal structure is known to us, and is based on subject matter knowledge. For example, we know that rain cases wet lawns but wet lawns don't cause rain.

Example 5 You may have noticed a correlation between rain and having a wet lawn, that is, the variable "Rain" is not independent of the variable "Wet Lawn" and hence $p_{R,W}(r, w) \neq$

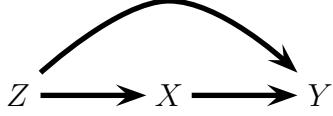


Figure 2: A basic causal graph. The arrows represent the effect of interventions. For example, the arrow from X to Y means that changing X effects the distribution of Y .

$p_R(r)p_W(w)$ where R denotes Rain and W denotes Wet Lawn. Consider the following two DAGs:

$$\text{Rain} \longrightarrow \text{Wet Lawn} \quad \text{Rain} \longleftarrow \text{Wet Lawn}.$$

The first DAG implies that $p(w, r) = p(r)p(w|r)$ while the second implies that $p(w, r) = p(w)p(r|w)$. No matter what the joint distribution $p(w, r)$ is, both graphs are correct. Both imply that R and W are not independent. But, intuitively, if we want a graph to indicate causation, the first graph is right and the second is wrong. Throwing water on your lawn doesn't cause rain. The reason we feel the first is correct while the second is wrong is because the interventions implied by the first graph are correct.

Look at the first graph and form the intervention $W = 1$ where 1 denotes "wet lawn." Following the rules of intervention, we break the arrows into W to get the modified graph:

$$\text{Rain} \quad \boxed{\text{set } \text{Wet Lawn} = 1}$$

with distribution $p^*(r) = p(r)$. Thus $\mathbb{P}(R = r | W := w) = \mathbb{P}(R = r)$ tells us that "wet lawn" does not cause rain.

Suppose we (wrongly) assume that the second graph is the correct causal graph and form the intervention $W = 1$ on the second graph. There are no arrows into W that need to be broken so the intervention graph is the same as the original graph. Thus $p^*(r) = p(r|w)$ which would imply that changing "wet" changes "rain." Clearly, this is nonsense.

Both are correct probability graphs but only the first is correct causally. We know the correct causal graph by using background knowledge.

Causal graphs can also be represented by structural equation models. The graph in Figure 2 can be written as:

$$\begin{aligned} Z &= g_1(U) \\ X &= g_2(Z, V) \\ Y &= g_3(Z, X, W) \end{aligned}$$

for some functions g_1, g_2, g_3 and some random variables (U, V, W) . Intervening on X corresponds to replacing the second equation with

$$X = x.$$

4 Causal Discovery Is Impossible

We could try to learn the correct causal graph from data but this is dangerous. In fact it is impossible. There are claims that it is possible but these claims are based on some unusual and not very convincing asymptotics. Specifically, there are claims that the graph can be discovered with some procedure and that the procedure is correct with probability tending to 1 as $n \rightarrow \infty$. But the asymptotic statement is non-standard: there is no finite sample size, however large, that can ever approximate the infinite limit.

What's worse, if we try to form a confidence interval for the size of the causal effect, then the confidence is infinite no matter how large the sample is. This is Panglossian asymptotics. To understand what is going on, let's consider two examples.

Suppose we observe $(X_1, Y_1), \dots, (X_n, Y_n)$ where X_i is the income of the subject's parents when the subject was a child, and Y_i is income of the subject at age 50. In this case, the variables are time ordered. So we can have X causing Y but we cannot have Y causing X . We must **always** allow for the fact that there may be many unobserved confounding variables. We will denote these by $U = (U_1, \dots, U_k)$ where k is potentially very large. There are eight possible graphs as shown in Figure 3.² Our main interest is in whether there is an arrow from X to Y .

Let's see how the graph discovery community reasons in this case. Suppose we observe a large sample $(X_1, Y_1), \dots, (X_n, Y_n)$. Let α be some measure of dependence between X and Y . It is possible to define a consistent estimator $\hat{\alpha}$. The causal discovery algorithms work as follows in this example. Suppose we find that there is a strong association between X and Y . (We can formally test for dependence between X and Y .) This is consistent with graphs 4,5,6,7 and 8. Some of these graphs include an arrow from X to Y and some don't. The conclusion is that we cannot tell if X causes Y . In this case, the causal discovery algorithms are correct.

Now suppose instead that we find that there is no significant association between X and Y . This is consistent with the first three graphs. None of these graphs include an arrow from X to Y . However, the last graph is also consistent with X being independent of Y . This might seem counterintuitive when you look at this graph. But the correlation created by the path $U \rightarrow X \rightarrow Y$ can cancel out the correlation created by the path $U \rightarrow Y$. Such a cancellation is called *unfaithfulness*. Such a cancellation is considered to be unlikely. And the set \mathcal{B} of such unfaithful distributions is “small.” (For example, if the joint distribution is Normal, then the parameters that correspond to unfaithful distributions have measure 0.) So it seems reasonable to restrict ourselves to faithful distributions. If we restrict to faithful distributions, then the only explanation for the independence of X and Y is the first three graphs. We conclude that X does not cause Y .

Let me summarize the logic. There is a measure of dependence α and a consistency estimator $\hat{\alpha}$. We are interested in the causal effect θ . We showed earlier that θ is a function $p(x, y, u)$. In particular, $\theta = 0$ means there is no arrow from X to Y and $\theta \neq 0$ means there

²Actually, we should have a separate node for each U_j . And then there are many more possible graphs.

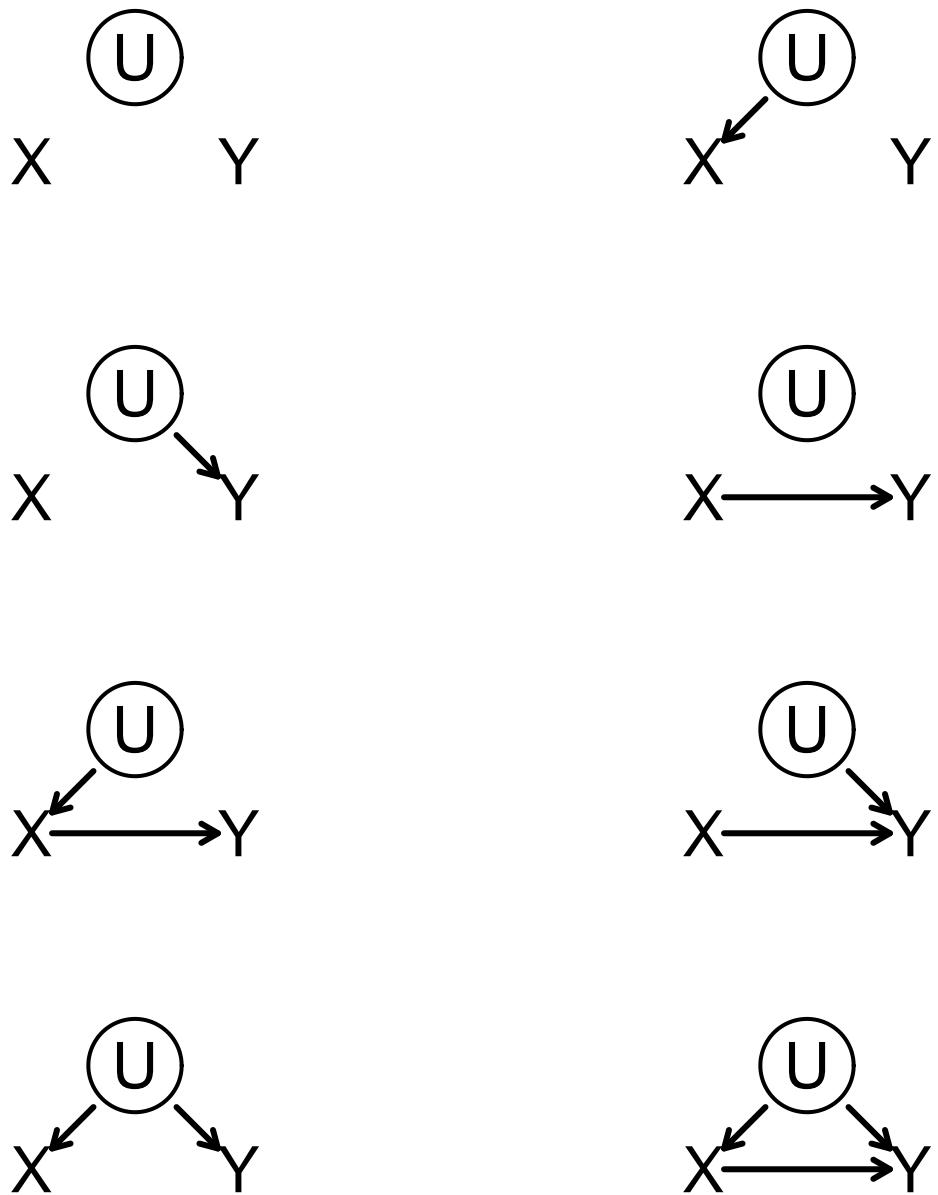


Figure 3: The eight possible causal graphs corresponding to the example.

is an arrow from X to Y . We have:

$$\begin{aligned}\alpha \neq 0 &\implies \theta \text{ can be } 0 \text{ or nonzero (no conclusion)} \\ \alpha = 0 \text{ and faithfulness} &\implies \theta = 0 \text{ (no causal effect).}\end{aligned}$$

Since $\hat{\alpha}$ is a consistent estimator of α , we can substitute $\hat{\alpha}$ for α and our conclusion is asymptotically correct. Note that if $P \in \mathcal{B}$, the relationship between α and θ breaks down. If $P \in \mathcal{B}$ then $\theta \neq 0$ but $\alpha = 0$.

Unfortunately, this reasoning is invalid. Let \mathcal{P} be a set of distributions for (X, Y, U) . Our model is

$$\mathcal{P}' = \mathcal{P} - \mathcal{B}$$

where \mathcal{B} is the set of unfaithful distributions. The problem is that we can explain $\hat{\alpha} \approx 0$ by graph 1 or by a P that is close to \mathcal{B} . We can always find a distribution P that is faithful but arbitrarily close to unfaithful. We can never tell if $\hat{\alpha} \approx 0$ is due to “no arrow from X to Y ” or from P being very close to unfaithful. No matter how large n is, we can find a P that is so close to unfaithful that it could result in $\hat{\alpha} \approx 0$.

By the way, keep in mind that U is very high dimensional. The set \mathcal{B} might be “small” in some sense, but it is very complex. It is like a spider web.

To simplify matters, consider the linear case. The model for the DAG is

$$\begin{aligned}U &= \epsilon_1 \\ X &= aU + \epsilon_2 \\ Y &= bX + cU + \epsilon_3.\end{aligned}$$

Here, the ϵ_i 's are mean 0 error terms. The causal effect is b . But all we observe is (X, Y) . The correlation between X and Y is $\rho = a^2 + ac + b$. The problem is:

It is easy to construct cases where b is huge but $\rho \approx 0$. Ruling out the case when b is large and $\rho = 0$ (unfaithfulness) isn't enough but we can still have b large and $\rho \approx 0$.

To make all this more precise, let $\psi = 1$ if there is an arrow from X to Y and let $\psi = 0$ if there is no arrow from X to Y . Let $\hat{\psi}$ be the output of any causal discovery procedure (which can be set-valued). Suppose that $\hat{\psi}$ is non-trivial, meaning that $1 \in \hat{\psi}$ with increasing probability when $b \neq 0$. Let \mathcal{P}_0 be the set of faithful distributions with zero causal effect.

Theorem 6 *For any non-trivial procedure,*

$$\sup_{P \in \mathcal{P}_0} P(\hat{\psi} = \psi) \rightarrow 1$$

as $n \rightarrow \infty$. In other words, if the procedure is non-trivial, we cannot control the type I error.

This result follows since there are infinitely many distributions in \mathcal{P}' that are arbitrarily close to \mathcal{B} and the procedure breaks down at \mathcal{B} . **The problem is that asymptotics have to be uniform over \mathcal{P} .** This is a point I have emphasized many times in this course. Uniformity is critical for sound statistical reasoning.

There is another way to see the problem. Consider the causal effect

$$\theta(x) = E[Y(x)] = \int \mathbb{E}[Y|X = x, U = u]p(u)du = \int m(x, u)p(u)du.$$

Discovering the graph involves implicitly estimating (or testing) $\theta(x)$. But it is clear that $\theta(x)$ is not estimable. It depends on $\mathbb{E}[Y|X = x, U = u]$ and $p(u)$. But we never observe U . **We can't estimate $m(x, u)$ if we don't observe u . Hence we can't estimate the causal effect.** We can't estimate parameters that are functions of unobserved random variables! The causal parameter is not identified. It is easy to show that the only valid confidence interval for $\theta(x)$ is the entire real line. In other words, if we want

$$\liminf_{n \rightarrow \infty} \inf_{P \in \mathcal{P}'} P(\theta(x) \in C_n) \geq 1 - \alpha$$

then $C_n = \mathbb{R}$ with high probability. This shows that the causal effect cannot be estimated.

For yet another perspective, let us suppose that we model the whole distribution. The distribution is

$$p(u, x, y) = p(u)p(x|u)p(y|x, u) = p(u_1, \dots, u_k)p(x|u_1, \dots, u_k)p(y|x, u_1, \dots, u_k).$$

The unknown parameters are the three functions $p(u_1, \dots, u_k)$, $p(x|u_1, \dots, u_k)$ and $p(y|x, u_1, \dots, u_k)$. Suppose we take a chance and assume these distributions are Normal. We can then get the mle for the parameters and hence for $\theta(x)$. But again, we don't observe any U 's. It's easy to see that the mle is for $\theta(x)$ is not defined. That is, every value of $\theta(x)$ is an mle.

To have reliable inference we need uniformly consistent estimates and we need valid confidence sets. There are no consistent estimators or valid confidence sets for causal parameters when there is unobserved confounding. The only solutions are: measure the confounders or do a randomized study.

Things get even worse when there are more than two variables. Let's consider another example. Suppose that we have the time ordered variables X, Y, Z . There are potential (unobserved) confounders U and V . See Figure 4. Again, the causal effects are not identified. There is nothing we can do here. But let's follow the causal discovery logic.

Suppose we observe a large sample and find that (i) X and Y are dependent, (ii) Y and Z are dependent and (iii) X and Z are conditionally independent given Y . I will explain in class how we can use the logic of causal discovery to conclude that:

- (a) X causes Y
- (b) Y causes Z
- (c) there are no confounding variables in the Universe.

The last conclusion is astounding and should be a hint that something is wrong.

Summary: Here is the bottom line:

1. In any real example based on observational data, we have to allow for the possibility that there are unobserved confounding variables.

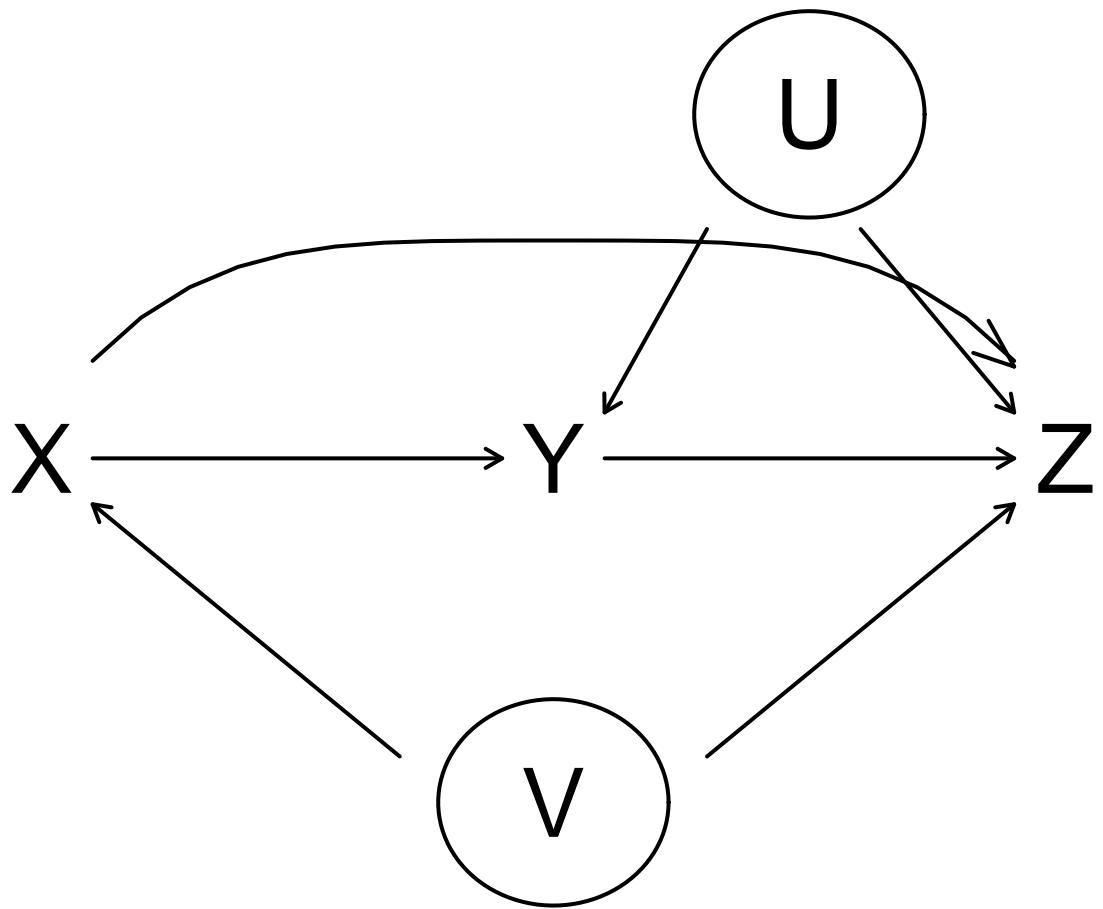


Figure 4: The full causal graph for the second example.

2. Causal quantities are functions of these unobserved variables.
3. It is impossible to estimate anything that is a function of unobserved variables.
4. Therefore, causal discovery is impossible.

Further Reading: A good tutorial with lots of good references is:

E. Kennedy (2015). Semiparametric Theory and Empirical Processes in Causal Inference.
arXiv:1510.04740

Also, there is a very good, free book here:

<https://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/>

Appendix: More on Graphical Interventions

If you are having difficulty understanding the difference between $p(y|x)$ and $p(y|\text{set } x)$, then this section will provide additional explanation. It is helpful to consider two different computer programs. Consider the DAG in Figure 2. The probability function for a distribution consistent with this DAG has the form $p(x, y, z) = p(x)p(y|x)p(z|x, y)$. The following is pseudocode for generating from this distribution.

For $i = 1, \dots, n$:

$$\begin{aligned} x_i &\leftarrow p_X(x_i) \\ y_i &\leftarrow p_{Y|X}(y_i|x_i) \\ z_i &\leftarrow p_{Z|X,Y}(z_i|x_i, y_i) \end{aligned}$$

Suppose we run this code, yielding data $(x_1, y_1, z_1), \dots, (x_n, y_n, z_n)$. Among all the times that we observe $Y = y$, how often is $Z = z$? The answer to this question is given by the conditional distribution of $Z|Y$. Specifically,

$$\begin{aligned} \mathbb{P}(Z = z|Y = y) &= \frac{\mathbb{P}(Y = y, Z = z)}{\mathbb{P}(Y = y)} = \frac{p(y, z)}{p(y)} \\ &= \frac{\sum_x p(x, y, z)}{p(y)} = \frac{\sum_x p(x)p(y|x)p(z|x, y)}{p(y)} \\ &= \sum_x p(z|x, y) \frac{p(y|x)p(x)}{p(y)} = \sum_x p(z|x, y) \frac{p(x, y)}{p(y)} \\ &= \sum_x p(z|x, y)p(x|y). \end{aligned}$$

Now suppose we **intervene** by changing the computer code. Specifically, suppose we fix Y at the value y . The code now looks like this:

```

set  $Y = y$ 
for  $i = 1, \dots, n$ 
 $x_i \leftarrow p_X(x_i)$ 
 $z_i \leftarrow p_{Z|X,Y}(z_i|x_i, y)$ 

```

Having $\text{set } Y = y$, how often was $Z = z$? To answer, note that the intervention has changed the joint probability to be

$$p^*(x, z) = p(x)p(z|x, y).$$

The answer to our question is given by the marginal distribution

$$p^*(z) = \sum_x p^*(x, z) = \sum_x p(x)p(z|x, y).$$

This is $p(z|\text{set } Y = y)$.

Minimax Theory

1 Introduction

When solving a statistical learning problem, there are often many procedures to choose from. This leads to the following question: how can we tell if one statistical learning procedure is better than another? One answer is provided by *minimax theory* which is a set of techniques for finding the minimum, worst case behavior of a procedure.

2 Definitions and Notation

Let \mathcal{P} be a set of distributions and let X_1, \dots, X_n be a sample from some distribution $P \in \mathcal{P}$. Let $\theta(P)$ be some function of P . For example, $\theta(P)$ could be the mean of P , the variance of P or the density of P . Let $\hat{\theta} = \hat{\theta}(X_1, \dots, X_n)$ denote an estimator. Given a metric d , the *minimax risk* is

$$R_n \equiv R_n(\mathcal{P}) = \inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \quad (1)$$

where the infimum is over all estimators. The *sample complexity* is

$$n(\epsilon, \mathcal{P}) = \min \left\{ n : R_n(\mathcal{P}) \leq \epsilon \right\}. \quad (2)$$

Example 1 Suppose that $\mathcal{P} = \{N(\theta, 1) : \theta \in \mathbb{R}\}$ where $N(\theta, 1)$ denotes a Gaussian with mean θ and variance 1. Consider estimating θ with the metric $d(a, b) = (a - b)^2$. The minimax risk is

$$R_n = \inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[(\hat{\theta} - \theta)^2]. \quad (3)$$

In this example, θ is a scalar.

Example 2 Let $(X_1, Y_1), \dots, (X_n, Y_n)$ be a sample from a distribution P . Let $m(x) = \mathbb{E}_P(Y|X = x) = \int y dP(y|X = x)$ be the regression function. In this case, we might use the metric $d(m_1, m_2) = \int (m_1(x) - m_2(x))^2 dx$ in which case the minimax risk is

$$R_n = \inf_{\hat{m}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \left[\int (\hat{m}(x) - m(x))^2 \right]. \quad (4)$$

In this example, θ is a function.

Notation. Recall that the *Kullback-Leibler distance* between two distributions P_0 and P_1 with densities p_0 and p_1 is defined to be

$$\text{KL}(P_0, P_1) = \int \log\left(\frac{dP_0}{dP_1}\right) dP_0 \int \log\left(\frac{p_0(x)}{p_1(x)}\right) p_0(x) dx.$$

The appendix defines several other distances between probability distributions and explains how these distances are related. We write $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. If P is a distribution with density p , the product distribution for n iid observations is P^n with density $p^n(x) = \prod_{i=1}^n p(x_i)$. It is easy to check that $\text{KL}(P_0^n, P_1^n) = n\text{KL}(P_0, P_1)$. For positive sequences a_n and b_n we write $a_n = \Omega(b_n)$ to mean that there exists $C > 0$ such that $a_n \geq Cb_n$ for all large n . $a_n \asymp b_n$ if a_n/b_n is strictly bounded away from zero and infinity for all large n ; that is, $a_n = O(b_n)$ and $b_n = O(a_n)$.

3 Bounding the Minimax Risk

Usually, we do not find R_n directly. Instead, we find an upper bound U_n and a lower bound L_n on R_n . To find an upper bound, let $\hat{\theta}$ be any estimator. Then

$$R_n = \inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \leq \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \equiv U_n. \quad (5)$$

So the maximum risk of any estimator provides an upper bound U_n . Finding a lower bound L_n is harder. We will consider three methods: the *Le Cam method*, the *Fano method* and *Tsybakov's bound*. If the lower and upper bound are close, then we have succeeded. For example, if $L_n = cn^{-\alpha}$ and $U_n = Cn^{-\alpha}$ for some positive constants c, C and α , then we have established that the *minimax rate of convergence* is $n^{-\alpha}$.

All the lower bound methods involve a the following trick: we reduce the problem to a hypothesis testing problem. It works like this. First, we will choose a finite set of distributions $M = \{P_1, \dots, P_N\} \subset \mathcal{P}$. Then

$$R_n = \inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \inf_{\hat{\theta}} \max_{P_j \in M} \mathbb{E}_j[d(\hat{\theta}, \theta_j)] \quad (6)$$

where $\theta_j = \theta(P_j)$ and \mathbb{E}_j is the expectation under P_j . Let $s = \min_{j \neq k} d(\theta_j, \theta_k)$. By Markov's inequality,

$$P(d(\hat{\theta}, \theta) > t) \leq \frac{\mathbb{E}[d(\hat{\theta}, \theta)]}{t}$$

and so

$$\mathbb{E}[d(\hat{\theta}, \theta)] \geq tP(d(\hat{\theta}, \theta) > t).$$

Setting $t = s/2$, and using (6), we have

$$R_n \geq \frac{s}{2} \inf_{\hat{\theta}} \max_{P_j \in M} P_j(d(\hat{\theta}, \theta_j) > s/2). \quad (7)$$

Given any estimator $\hat{\theta}$, define

$$\psi^* = \operatorname{argmin}_j d(\hat{\theta}, \theta_j).$$

Now, if $\psi^* \neq j$ then, letting $k = \psi^*$,

$$\begin{aligned} s &\leq d(\theta_j, \theta_k) \leq d(\theta_j, \hat{\theta}) + d(\theta_k, \hat{\theta}) \\ &\leq d(\theta_j, \hat{\theta}) + d(\theta_j, \hat{\theta}) \text{ since } \psi^* \neq j \text{ implies that } d(\hat{\theta}, \theta_k) \leq d(\hat{\theta}, \theta_j) \\ &= 2d(\theta_j, \hat{\theta}). \end{aligned}$$

So $\psi^* \neq j$ implies that $d(\theta_j, \hat{\theta}) \geq s/2$. Thus

$$P_j(d(\hat{\theta}, \theta_j) > s/2) \geq P_j(\psi^* \neq j) \geq \inf_{\psi} P_j(\psi \neq j)$$

where the infimum is over all maps ψ from the data to $\{1, \dots, N\}$. (We can think of ψ is a multiple hypothesis test.) Thus we have

$$R_n \geq \frac{s}{2} \inf_{\psi} \max_{P_j \in M} P_j(\psi \neq j).$$

We can summarize this as a theorem:

Theorem 3 Let $M = \{P_1, \dots, P_N\} \subset \mathcal{P}$ and let $s = \min_{j \neq k} d(\theta_j, \theta_k)$. Then

$$R_n = \inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{2} \inf_{\psi} \max_{P_j \in M} P_j(\psi \neq j). \quad (8)$$

Getting a good lower bound involves carefully selecting $M = \{P_1, \dots, P_N\}$. If M is too big, s will be small. If M is too small, then $\max_{P_j \in M} P_j(\psi \neq j)$ will be small.

4 Distances

We will need some distances between distributions. Specifically,

Total Variation	$\text{TV}(P, Q)$	$= \sup_A P(A) - Q(A) $
L_1	$\ P - Q\ _1$	$= \int p - q $
Kullback-Leibler	$\text{KL}(P, Q)$	$= \int p \log(p/q)$
χ^2	$\chi^2(P, Q)$	$= \int \left(\frac{p}{q} - 1\right)^2 dQ = \int \frac{p^2}{q} - 1$
Hellinger	$\text{H}(P, Q)$	$= \sqrt{\int (\sqrt{p} - \sqrt{q})^2}.$

We also define the *affinity* between P and q by

$$a(p, q) = \int (p \wedge q).$$

We then have

$$\text{TV}(P, Q) = \frac{1}{2} \|P - Q\|_1 = 1 - a(p, q)$$

and

$$\frac{1}{2} H^2(P, Q) \leq \text{TV}(P, Q) \leq \sqrt{\text{KL}(P, Q)} \leq \sqrt{\chi^2(P, Q)}.$$

The appendix contains more information about these distances.

5 Lower Bound Method 1: Le Cam

Theorem 4 Let \mathcal{P} be a set of distributions. For any pair $P_0, P_1 \in \mathcal{P}$,

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{4} \int [p_0^n(x) \wedge p_1^n(x)] dx = \frac{s}{4} [1 - \text{TV}(P_0^n, P_1^n)] \quad (9)$$

where $s = d(\theta(P_0), \theta(P_1))$. We also have:

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{8} e^{-n\text{KL}(P_0, P_1)} \geq \frac{s}{8} e^{-n\chi^2(P_0, P_1)} \quad (10)$$

and

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{8} \left(1 - \frac{1}{2} \int |p_0 - p_1|\right)^{2n}. \quad (11)$$

Corollary 5 Suppose there exist $P_0, P_1 \in \mathcal{P}$ such that $\text{KL}(P_0, P_1) \leq \log 2/n$. Then

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{16} \quad (12)$$

where $s = d(\theta(P_0), \theta(P_1))$.

Proof. Let $\theta_0 = \theta(P_0)$, $\theta_1 = \theta(P_1)$ and $s = d(\theta_0, \theta_1)$. First suppose that $n = 1$ so that we have a single observation X . From Theorem 3,

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{2}\pi$$

where

$$\pi = \inf_{\psi} \max_{j=0,1} P_j(\psi \neq j). \quad (13)$$

Since a maximum is larger than an average,

$$\pi = \inf_{\psi} \max_{j=0,1} P_j(\psi \neq j) \geq \inf_{\psi} \frac{P_0(\psi \neq 0) + P_1(\psi \neq 1)}{2}.$$

Define the *Neyman-Pearson test*

$$\psi_*(x) = \begin{cases} 0 & \text{if } p_0(x) \geq p_1(x) \\ 1 & \text{if } p_0(x) < p_1(x). \end{cases}$$

In Lemma 7 below, we show that the sum of the errors $P_0(\psi \neq 0) + P_1(\psi \neq 1)$ is minimized by ψ^* . Now

$$\begin{aligned} P_0(\psi_* \neq 0) + P_1(\psi_* \neq 1) &= \int_{p_1 > p_0} p_0(x) dx + \int_{p_0 > p_1} p_1(x) dx \\ &= \int_{p_1 > p_0} [p_0(x) \wedge p_1(x)] dx + \int_{p_0 > p_1} [p_0(x) \wedge p_1(x)] dx = \int [p_0(x) \wedge p_1(x)] dx. \end{aligned}$$

Thus,

$$\frac{P_0(\psi_* \neq 0) + P_1(\psi_* \neq 1)}{2} = \frac{1}{2} \int [p_0(x) \wedge p_1(x)] dx.$$

Thus we have shown that

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{4} \int [p_0(x) \wedge p_1(x)] dx.$$

Now suppose we have n observations. Then, replacing p_0 and p_1 with $p_0^n(x) = \prod_{i=1}^n p_0(x_i)$ and $p_1^n(x) = \prod_{i=1}^n p_1(x_i)$, we have

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{4} \int [p_0^n(x) \wedge p_1^n(x)] dx.$$

In Lemma 7 below, we show that $\int p \wedge q \geq \frac{1}{2} e^{-\mathbf{KL}(P, Q)}$. Since $\mathbf{KL}(P_0^n, P_1^n) = n \mathbf{KL}(P_0, P_1)$, we have

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{8} e^{-n \mathbf{KL}(P_0, P_1)}.$$

The other results follow from the inequalities on the distances. \square

Lemma 6 *Let ψ^* be the Neyman-Pearson test. For any test ψ ,*

$$P_0(\psi = 1) + P_1(\psi = 0) \geq P_0(\psi^* = 1) + P_1(\psi^* = 0).$$

Proof. Recall that $p_0 > p_1$ when $\psi^* = 0$ and that $p_0 < p_1$ when $\psi^* = 1$. So

$$\begin{aligned}
P_0(\psi = 1) + P_1(\psi = 0) &= \int_{\psi=1} p_0(x)dx + \int_{\psi=0} p_1(x)dx \\
&= \int_{\psi=1, \psi^*=1} p_0(x)dx + \int_{\psi=1, \psi^*=0} p_0(x)dx + \int_{\psi=0, \psi^*=0} p_1(x)dx + \int_{\psi=0, \psi^*=1} p_1(x)dx \\
&\geq \int_{\psi=1, \psi^*=1} p_0(x)dx + \int_{\psi=1, \psi^*=0} p_1(x)dx + \int_{\psi=0, \psi^*=0} p_1(x)dx + \int_{\psi=0, \psi^*=1} p_0(x)dx \\
&= \int_{\psi^*=1} p_0(x)dx + \int_{\psi^*=0} p_1(x)dx \\
&= P_0(\psi^* = 1) + P_1(\psi^* = 0).
\end{aligned}$$

□

Lemma 7 For any P and Q , $\int p \wedge q \geq \frac{1}{2}e^{-\text{KL}(P,Q)}$.

Proof. First note that, since $(a \vee b) + (a \wedge b) = a + b$, we have

$$\int(p \vee q) + \int(p \wedge q) = 2. \quad (14)$$

Hence

$$\begin{aligned}
2 \int p \wedge q &\geq 2 \int p \wedge q - \left(\int p \wedge q \right)^2 = \left(\int p \wedge q \right) \left[2 - \int p \wedge q \right] \\
&= \left(\int p \wedge q \right) \left(\int p \vee q \right) \text{ from (14)} \\
&\geq \left(\int \sqrt{(p \wedge q)(p \vee q)} \right)^2 \text{ Cauchy - Schwartz} \\
&= \left(\int \sqrt{pq} \right)^2 = \exp \left(2 \log \int \sqrt{pq} \right) \\
&= \exp \left(2 \log \int p \sqrt{q/p} \right) \geq \exp \left(2 \int p \log \sqrt{\frac{q}{p}} \right) = e^{-\text{KL}(P,Q)}
\end{aligned}$$

where we used Jensen's inequality in the last inequality. □

Example 8 Consider data $(X_1, Y_1), \dots, (X_n, Y_n)$ where $X_i \sim \text{Uniform}(0, 1)$, $Y_i = m(X_i) + \epsilon_i$ and $\epsilon_i \sim N(0, 1)$. Assume that

$$m \in \mathcal{M} = \left\{ m : |m(y) - m(x)| \leq L|x - y|, \text{ for all } x, y \in [0, 1] \right\}.$$

So \mathcal{P} is the set of distributions of the form $p(x, y) = p(x)p(y|x) = \phi(y - m(x))$ where $m \in \mathcal{M}$.

How well can we estimate $m(x)$ at some point x ? Without loss of generality, let's take $x = 0$ so the parameter of interest is $\theta = m(0)$. Let $d(\theta_0, \theta_1) = |\theta_0 - \theta_1|$. Let $m_0(x) = 0$ for all x . Let $0 \leq \epsilon \leq 1$ and define

$$m_1(x) = \begin{cases} L(\epsilon - x) & 0 \leq x \leq \epsilon \\ 0 & x \geq \epsilon. \end{cases}$$

Then $m_0, m_1 \in \mathcal{M}$ and $s = |m_1(0) - m_0(0)| = L\epsilon$. The KL distance is

$$\begin{aligned} \text{KL}(P_0, P_1) &= \int_0^1 \int p_0(x, y) \log \left(\frac{p_0(x, y)}{p_1(x, y)} \right) dy dx \\ &= \int_0^1 \int p_0(x) p_0(y|x) \log \left(\frac{p_0(x) p_0(y|x)}{p_1(x) p_1(y|x)} \right) dy dx \\ &= \int_0^1 \int \phi(y) \log \left(\frac{\phi(y)}{\phi(y - m_1(x))} \right) dy dx \\ &= \int_0^\epsilon \int \phi(y) \log \left(\frac{\phi(y)}{\phi(y - m_1(x))} \right) dy dx \\ &= \int_0^\epsilon \text{KL}(N(0, 1), N(m_1(x), 1)) dx. \end{aligned}$$

Now, $\text{KL}(N(\mu_1, 1), N(\mu_2, 1)) = (\mu_1 - \mu_2)^2/2$. So

$$\text{KL}(P_0, P_1) = \frac{L^2}{2} \int_0^\epsilon (\epsilon - x)^2 dx = \frac{L^2 \epsilon^3}{6}.$$

Let $\epsilon = (6 \log 2 / (L^2 n))^{1/3}$. Then, $\text{KL}(P_0, P_1) = \log 2/n$ and hence, by Corollary 5,

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{16} = \frac{L\epsilon}{16} = \frac{L}{16} \left(\frac{6 \log 2}{L^2 n} \right)^{1/3} = \left(\frac{c}{n} \right)^{1/3}. \quad (15)$$

It is easy to show that the regressogram (regression histogram) $\hat{\theta} = \hat{m}(0)$ has risk

$$\mathbb{E}_P[d(\hat{\theta}, \theta(P))] \leq \left(\frac{C}{n} \right)^{1/3}.$$

Thus we have proved that

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \asymp n^{-\frac{1}{3}}. \quad (16)$$

The same calculations in d dimensions yield

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \asymp n^{-\frac{1}{d+2}}. \quad (17)$$

On the squared scale we have

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d^2(\hat{\theta}, \theta(P))] \asymp n^{-\frac{2}{d+2}}. \quad (18)$$

Similar rates hold in density estimation.

There is a more general version of Le Cam's lemma that is sometimes useful.

Lemma 9 Let P, Q_1, \dots, Q_N be distributions such that $d(\theta(P), \theta(Q_j)) \geq s$ for all j . Then

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{4} \int (p^n \wedge q^n)$$

where $q = \frac{1}{N} \sum_j q_j$.

Example 10 Let

$$Y_i = \theta_i + \frac{1}{\sqrt{n}} Z_i, \quad i = 1, \dots, d$$

where $Z_1, Z_2, \dots, Z_d \sim N(0, 1)$ and $\theta = (\theta_1, \dots, \theta_d) \in \Theta$ where $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq 1\}$. Let $P = N(0, n^{-1}I)$. Let Q_j have mean 0 expect that j^{th} coordinate has mean $\sqrt{a \log d/n}$ where $0 < a < 1$. Let $q = \frac{1}{N} \sum_j q_j$. Some algebra (good homework question!) shows that $\chi^2(q, p) \rightarrow 0$ as $d \rightarrow \infty$. By the generalized Le Cam lemma, $R_n \geq a \log d/n$ using squared error loss. We can estimate θ by thresholding (Bonferroni). This gives a matching upper bound.

6 Lower Bound Method II: Fano

For metrics like $d(f, g) = \int (f - g)^2$, Le Cam's method will usually not give a tight bound. Instead, we use Fano's method. Instead of choosing two distributions P_0, P_1 , we choose a finite set of distributions $P_1, \dots, P_N \in \mathcal{P}$.

We start with Fano's lemma.

Lemma 11 (Fano) Let $X_1, \dots, X_n \sim P$ where $P \in \{P_1, \dots, P_N\}$. Let ψ be any function of X_1, \dots, X_n taking values in $\{1, \dots, N\}$. Let $\beta = \max_{j \neq k} \text{KL}(P_j, P_k)$. Then

$$\frac{1}{N} \sum_{j=1}^N P_j(\psi \neq j) \geq \left(1 - \frac{n\beta + \log 2}{\log N}\right).$$

Proof: See Lemma 28 in the appendix. \square .

Now we can state and prove the Fano minimax bound.

Theorem 12 *Let $F = \{P_1, \dots, P_N\} \subset \mathcal{P}$. Let $\theta(P)$ be a parameter taking values in a metric space with metric d . Then*

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \left(d(\hat{\theta}, \theta(P)) \right) \geq \frac{s}{2} \left(1 - \frac{n\beta + \log 2}{\log N} \right) \quad (19)$$

where

$$s = \min_{j \neq k} d(\theta(P_j), \theta(P_k)), \quad (20)$$

and

$$\beta = \max_{j \neq k} \text{KL}(P_j, P_k). \quad (21)$$

Corollary 13 (Fano Minimax Bound) *Suppose there exists $F = \{P_1, \dots, P_N\} \subset \mathcal{P}$ such that $N \geq 16$ and*

$$\beta = \max_{j \neq k} \text{KL}(P_j, P_k) \leq \frac{\log N}{4n}. \quad (22)$$

Then

$$\inf_{\hat{\theta}} \max_{P \in \mathcal{P}} \mathbb{E}_P \left[d(\hat{\theta}, \theta(P)) \right] \geq \frac{s}{4}. \quad (23)$$

Proof. From Theorem 3,

$$R_n \geq \frac{s}{2} \inf_{\psi} \max_{P_j \in F} P_j(\psi \neq j) \geq \frac{s}{2} \frac{1}{N} \sum_{j=1}^N P_j(\psi \neq j)$$

where the latter is due to the fact that a max is larger than an average. By Fano's lemma,

$$\frac{1}{N} \sum_{j=1}^N P_j(\psi \neq j) \geq \left(1 - \frac{n\beta + \log 2}{\log N} \right).$$

Thus,

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \left(d(\hat{\theta}, \theta(P)) \right) \geq \inf_{\hat{\theta}} \max_{P \in \mathcal{F}} \mathbb{E}_P \left(d(\hat{\theta}, \theta(P)) \right) \geq \frac{s}{2} \left(1 - \frac{n\beta + \log 2}{\log N} \right). \quad (24)$$

\square

7 Lower Bound Method III: Tsybakov's Bound

This approach is due to Tsybakov (2009). The proof of the following theorem is in the appendix.

Theorem 14 (Tsybakov 2009) *Let $X_1, \dots, X_n \sim P \in \mathcal{P}$. Let $\{P_0, P_1, \dots, P_N\} \subset \mathcal{P}$ where $N \geq 3$. Assume that P_0 is absolutely continuous with respect to each P_j . Suppose that*

$$\frac{1}{N} \sum_{j=1}^N \text{KL}(P_j, P_0) \leq \frac{\log N}{16}.$$

Then

$$\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{\theta}, \theta(P))] \geq \frac{s}{16}$$

where

$$s = \max_{0 \leq j < k \leq N} d(\theta(P_j), \theta(P_k)).$$

8 Hypercubes

To use Fano's method or Tsybakov's method, we need to construct a finite class of distributions \mathcal{F} . Sometimes we use a set of the form

$$\mathcal{F} = \left\{ P_\omega : \omega \in \Omega \right\}$$

where

$$\Omega = \left\{ \omega = (\omega_1, \dots, \omega_m) : \omega_i \in \{0, 1\}, i = 1, \dots, m \right\}$$

which is called a hypercube. There are $N = 2^m$ distributions in \mathcal{F} . For $\omega, \nu \in \Omega$, define the *Hamming distance* $H(\omega, \nu) = \sum_{j=1}^m I(\omega_j \neq \nu_j)$.

One problem with a hypercube is that some pairs $P, Q \in \mathcal{F}$ might be very close together which will make $s = \min_{j \neq k} d(\theta(P_j), \theta(P_k))$ small. This will result in a poor lower bound. We can fix this problem by pruning the hypercube. That is, we can find a subset $\Omega' \subset \Omega$ which has nearly the same number of elements as Ω but such that each pair $P, Q \in \mathcal{F}' = \left\{ P_\omega : \omega \in \Omega' \right\}$ is far apart. We call Ω' a *pruned hypercube*. The technique for constructing Ω' is the *Varshamov-Gilbert lemma*.

Lemma 15 (Varshamov-Gilbert) *Let $\Omega = \left\{ \omega = (\omega_1, \dots, \omega_N) : \omega_j \in \{0, 1\} \right\}$. Suppose that $N \geq 8$. There exists $\omega^0, \omega^1, \dots, \omega^M \in \Omega$ such that (i) $\omega^0 = (0, \dots, 0)$, (ii) $M \geq 2^{N/8}$ and (iii) $H(\omega^{(j)}, \omega^{(k)}) \geq N/8$ for $0 \leq j < k \leq M$. We call $\Omega' = \{\omega^0, \omega^1, \dots, \omega^M\}$ a pruned hypercube.*

Proof. Let $D = \lfloor N/8 \rfloor$. Set $\omega^0 = (0, \dots, 0)$. Define $\Omega_0 = \Omega$ and $\Omega_1 = \{\omega \in \Omega : H(\omega, \omega^0) > D\}$. Let ω^1 be any element in Ω_1 . Thus we have eliminated $\{\omega \in \Omega : H(\omega, \omega^0) \leq D\}$. Continue this way recursively and at the j^{th} step define $\Omega_j = \{\omega \in \Omega_{j-1} : H(\omega, \omega^{j-1}) > D\}$ where $j = 1, \dots, M$. Let n_j be the number of elements eliminated at step j , that is, the number of elements in $A_j = \{\omega \in \Omega_j : H(\omega, \omega^{(j)}) \leq D\}$. It follows that

$$n_j \leq \sum_{i=0}^D \binom{N}{i}.$$

The sets A_0, \dots, A_M partition Ω and so $n_0 + n_1 + \dots + n_M = 2^N$. Thus,

$$(M+1) \sum_{i=0}^D \binom{N}{i} \geq 2^N.$$

Thus

$$M+1 \geq \frac{1}{\sum_{i=0}^D 2^{-N} \binom{N}{i}} = \frac{1}{\mathbb{P}\left(\sum_{i=1}^N Z_i \leq \lfloor m/8 \rfloor\right)}$$

where Z_1, \dots, Z_N are iid Bernoulli $(1/2)$ random variables. By Hoeffding's inequality,

$$\mathbb{P}\left(\sum_{i=1}^N Z_i \leq \lfloor m/8 \rfloor\right) \leq e^{-9N/32} < 2^{-N/4}.$$

Therefore, $M \geq 2^{N/8}$ as long as $N \geq 8$. Finally, note that, by construction, $H(\omega^j, \omega^k) \geq D+1 \geq N/8$. \square

Example 16 Consider data $(X_1, Y_1), \dots, (X_n, Y_n)$ where $X_i \sim \text{Uniform}(0, 1)$, $Y_i = f(X_i) + \epsilon_i$ and $\epsilon_i \sim N(0, 1)$. (The assumption that X is uniform is not crucial.) Assume that f is in the Holder class \mathcal{F} defined by

$$\mathcal{F} = \left\{ f : |f^{(\ell)}(y) - f^{(\ell)}(x)| \leq L|x - y|^{\beta - \ell}, \text{ for all } x, y \in [0, 1] \right\}$$

where $\ell = \lfloor \beta \rfloor$. \mathcal{P} is the set of distributions of the form $p(x, y) = p(x)p(y|x) = \phi(y - m(x))$ where $f \in \mathcal{F}$. Let Ω' be a pruned hypercube and let

$$\mathcal{F}' = \left\{ f_\omega(x) = \sum_{j=1}^m \omega_j \phi_j(x) : \omega \in \Omega' \right\}$$

where $m = \lceil cn^{\frac{1}{2\beta+1}} \rceil$, $\phi_j(x) = Lh^\beta K((x - X_j)/h)$, and $h = 1/m$. Here, K is any sufficiently smooth function supported on $(-1/2, 1/2)$. Let $d^2(f, g) = \int (f - g)^2$. Some calculations show that, for $\omega, \nu \in \Omega'$,

$$d(f_\omega, f_\nu) = \sqrt{H(\omega, \nu)} L h^{\beta + \frac{1}{2}} \int K^2 \geq \sqrt{\frac{m}{8}} L h^{\beta + \frac{1}{2}} \int K^2 \geq c_1 h^\beta.$$

We used the Varshamov-Gilbert result which implies that $H(\omega, \nu) \geq m/8$. Furthermore,

$$\text{KL}(P_\omega, P_\nu) \leq c_2 h^{2\beta}.$$

To apply Corollary 13, we need to have

$$\text{KL}(P_\omega, P_\nu) \leq \frac{\log N}{4n}.$$

Now

$$\frac{\log N}{4n} = \frac{\log 2^{m/8}}{4n} = \frac{m}{32n} = \frac{1}{32nh}.$$

So we set $h = (c/n)^{1/(2\beta+1)}$. In that case, $d(f_\omega, f_\nu) \geq c_1 h^\beta = c_1 (c/n)^{\beta/(2\beta+1)}$. Corollary 13 implies that

$$\inf_{\hat{f}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\hat{f}, f)] \geq n^{-\frac{\beta}{2\beta+1}}.$$

It follows that

$$\inf_{\hat{f}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \int (f - \hat{f})^2 \geq n^{-\frac{2\beta}{2\beta+1}}.$$

It can be shown that there are kernel estimators that achieve this rate of convergence. (The kernel has to be chosen carefully to take advantage of the degree of smoothness β .) A similar calculation in d dimensions shows that

$$\inf_{\hat{f}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \int (f - \hat{f})^2 \geq n^{-\frac{2\beta}{2\beta+d}}.$$

9 Further Examples

9.1 Parametric Maximum Likelihood

For parametric models that satisfy weak regularity conditions, the maximum likelihood estimator is approximately minimax. Consider squared error loss which is squared bias plus variance. In parametric models with large samples, it can be shown that the variance term dominates the bias so the risk of the mle $\hat{\theta}$ roughly equals the variance:¹

$$R(\theta, \hat{\theta}) = \text{Var}_\theta(\hat{\theta}) + \text{bias}^2 \approx \text{Var}_\theta(\hat{\theta}). \quad (25)$$

The variance of the mle is approximately $\text{Var}(\hat{\theta}) \approx \frac{1}{nI(\theta)}$ where $I(\theta)$ is the *Fisher information*. Hence,

$$nR(\theta, \hat{\theta}) \approx \frac{1}{I(\theta)}. \quad (26)$$

¹Typically, the squared bias is order $O(n^{-2})$ while the variance is of order $O(n^{-1})$.

For any other estimator θ' , it can be shown that for large n , $R(\theta, \theta') \geq R(\theta, \hat{\theta})$. For d -dimensional vectors we have $R(\theta, \hat{\theta}) \approx |I(\theta)|^{-1}/n = O(d/n)$.

Here is a more precise statement, due to Hájek and Le Cam. The family of distributions $(P_\theta : \theta \in \Theta)$ with densities $(P_\theta : \theta \in \Theta)$ is *differentiable in quadratic mean* if there exists ℓ'_θ such that

$$\int \left(\sqrt{p_{\theta+h}} - \sqrt{p_\theta} - \frac{1}{2} h^T \ell'_\theta \sqrt{p_\theta} \right)^2 d\mu = o(\|h\|^2). \quad (27)$$

Theorem 17 (Hájek and Le Cam) Suppose that $(P_\theta : \theta \in \Theta)$ is differentiable in quadratic mean where $\Theta \subset \mathbb{R}^k$ and that the Fisher information I_θ is nonsingular. Let ψ be differentiable. Then $\psi(\hat{\theta}_n)$, where $\hat{\theta}_n$ is the mle, is asymptotically, locally, uniformly minimax in the sense that, for any estimator T_n , and any bowl-shaped ℓ ,

$$\sup_{I \in \mathcal{I}} \liminf_{n \rightarrow \infty} \sup_{h \in I} \mathbb{E}_{\theta+h/\sqrt{n}} \ell \left(\sqrt{n} \left(T_n - \psi \left(\theta + \frac{h}{\sqrt{n}} \right) \right) \right) \geq \mathbb{E}(\ell(U)) \quad (28)$$

where \mathcal{I} is the class of all finite subsets of \mathbb{R}^k and $U \sim N(0, \psi'_\theta I_\theta^{-1} (\psi'_\theta)^T)$.

For a proof, see van der Vaart (1998). Note that the right hand side of the displayed formula is the risk of the mle. In summary: in well-behaved parametric models, with large samples, the mle is approximately minimax.

9.2 Estimating a Smooth Density

Here we use the general strategy to derive the minimax rate of convergence for estimating a smooth density. (See Yu (2008) for more details.)

Let \mathcal{F} be all probability densities f on $[0, 1]$ such that

$$0 < c_0 \leq f(x) \leq c_1 < \infty, \quad |f''(x)| \leq c_2 < \infty.$$

We observe $X_1, \dots, X_n \sim P$ where P has density $f \in \mathcal{F}$. We will use the squared Hellinger distance $d^2(f, g) = \int_0^1 (\sqrt{f(x)} - \sqrt{g(x)})^2 dx$ as a loss function.

Upper Bound. Let \hat{f}_n be the kernel estimator with bandwidth $h = n^{-1/5}$. Then, using bias-variance calculations, we have that

$$\sup_{f \in \mathcal{F}} \mathbb{E}_f \left(\int (\hat{f}(x) - f(x))^2 dx \right) \leq C n^{-4/5}$$

for some C . But

$$\int (\sqrt{f(x)} - \sqrt{g(x)})^2 dx = \int \left(\frac{f(x) - g(x)}{\sqrt{f(x)} + \sqrt{g(x)}} \right)^2 dx \leq C' \int (f(x) - g(x))^2 dx \quad (29)$$

for some C' . Hence $\sup_f \mathbb{E}_f(d^2(f, \hat{f}_n)) \leq Cn^{-4/5}$ which gives us an upper bound.

Lower Bound. For the lower bound we use Fano's inequality. Let g be a bounded, twice differentiable function on $[-1/2, 1/2]$ such that

$$\int_{-1/2}^{1/2} g(x) dx = 0, \int_{-1/2}^{1/2} g^2(x) dx = a > 0, \int_{-1/2}^{1/2} (g'(x))^2 dx = b > 0.$$

Fix an integer m and for $j = 1, \dots, m$ define $x_j = (j - (1/2))/m$ and

$$g_j(x) = \frac{c}{m^2} g(m(x - x_j))$$

for $x \in [0, 1]$ where c is a small positive constant. Let \mathcal{M} denote the Varshamov-Gilbert pruned version of the set

$$\left\{ f_\tau = 1 + \sum_{j=1}^m \tau_j g_j(x) : \tau = (\tau_1, \dots, \tau_m) \in \{-1, +1\}^m \right\}.$$

For $f_\tau \in \mathcal{M}$, let f_τ^n denote the product density for n observations and let $\mathcal{M}_n = \{f_\tau^n : f_\tau \in \mathcal{M}\}$. Some calculations show that, for all τ, τ' ,

$$\text{KL}(f_\tau^n, f_{\tau'}^n) = n \text{KL}(f_\tau, f_{\tau'}) \leq \frac{C_1 n}{m^4} \equiv \beta. \quad (30)$$

By Lemma 15, we can choose a subset F of \mathcal{M} with $N = e^{c_0 m}$ elements (where c_0 is a constant) and such that

$$d^2(f_\tau, f_{\tau'}) \geq \frac{C_2}{m^4} \equiv \alpha \quad (31)$$

for all pairs in F . Choosing $m = cn^{1/5}$ gives $\beta \leq \log N/4$ and $d^2(f_\tau, f_{\tau'}) \geq \frac{C_2}{n^{4/5}}$. Fano's lemma implies that

$$\max_j \mathbb{E}_j d^2(\hat{f}, f_j) \geq \frac{C}{n^{4/5}}.$$

Hence the minimax rate is $n^{-4/5}$ which is achieved by the kernel estimator. Thus we have shown that $R_n(\mathcal{P}) \asymp n^{-4/5}$.

This result can be generalized to higher dimensions and to more general measures of smoothness. Since the proof is similar to the one dimensional case, we state the result without proof.

Theorem 18 Let \mathcal{Z} be a compact subset of \mathbb{R}^d . Let $\mathcal{F}(p, C)$ denote all probability density functions on \mathcal{Z} such that

$$\int \sum \left| \frac{\partial^p}{\partial z_1^{p_1} \cdots \partial z_d^{p_d}} f(z) \right|^2 dz \leq C$$

where the sum is over all p_1, \dots, p_d such that $\sum_j p_j = p$. Then there exists a constant $D > 0$ such that

$$\inf_{\hat{f}} \sup_{f \in \mathcal{F}(p, C)} \mathbb{E}_f \int (\hat{f}_n(z) - f(z))^2 dz \geq D \left(\frac{1}{n} \right)^{\frac{2p}{2p+1}}. \quad (32)$$

The kernel estimator (with an appropriate kernel) with bandwidth $h_n = n^{-1/(2p+d)}$ achieves this rate of convergence.

9.3 Minimax Classification

Let us now turn to classification. We focus on some results of Yang (1999), Tsybakov (2004), Mammen and Tsybakov (1999), Audibert and Tsybakov (2005) and Tsybakov and van de Geer (2005).

The data are $Z = (X_1, Y_1), \dots, (X_n, Y_n)$ where $Y_i \in \{0, 1\}$. Recall that a classifier is a function of the form $h(x) = I(x \in G)$ for some set G . The classification risk is

$$R(G) = \mathbb{P}(Y \neq h(X)) = \mathbb{P}(Y \neq I(X \in G)) = \mathbb{E}(Y - I(X \in G))^2. \quad (33)$$

The optimal classifier is $h^*(x) = I(x \in G^*)$ where $G^* = \{x : m(x) \geq 1/2\}$ and $m(x) = \mathbb{E}(Y|X = x)$. We are interested in how close $R(G)$ is to $R(G^*)$. Following Tsybakov (2004) we define

$$d(G, G^*) = R(G) - R(G^*) = 2 \int_{G \Delta G^*} \left| m(x) - \frac{1}{2} \right| dP_X(x) \quad (34)$$

where $A \Delta B = (A \cap B^c) \cup (A^c \cup B)$ and P_X is the marginal distribution of X .

There are two common types of classifiers. The first type are *plug-in classifiers* of the form $\hat{h}(x) = I(\hat{m}(x) \geq 1/2)$ where \hat{m} is an estimate of the regression function. The second type are *empirical risk minimizers* where \hat{h} is taken to be the h that minimizes the observed error rate $n^{-1} \sum_{i=1}^n (Y_i \neq h(X_i))$ as h varies over a set of classifiers \mathcal{H} . Sometimes one minimizes the error rate plus a penalty term.

According to Yang (1999), the classification problem has, under weak conditions, the same order of difficulty (in terms of minimax rates) as estimating the regression function $m(x)$. Therefore the rates are given in Example 39. According to Tsybakov (2004) and Mammen and Tsybakov (1999), classification is easier than regression. The apparent discrepancy is due to differing assumptions.

To see that classification error cannot be harder than regression, note that for any \hat{m} and corresponding \hat{G}

$$d(G, \hat{G}) = 2 \int_{G \Delta \hat{G}} |m(x) - \frac{1}{2}| dP_X(x) \quad (35)$$

$$\leq 2 \int |\hat{m}(x) - m(x)| dP_X(x) \leq 2 \sqrt{\int (\hat{m}(x) - m(x))^2 dP_X(x)} \quad (36)$$

so the rate of convergence of $d(G, G^*)$ is at least as fast as the regression function.

Instead of putting assumptions on the regression function m , Mammen and Tsybakov (1999) put an entropy assumption on the set of *decision sets* \mathcal{G} . They assume

$$\log N(\epsilon, \mathcal{G}, d) \leq A\epsilon^{-\rho} \quad (37)$$

where $N(\epsilon, \mathcal{G}, d)$ is the smallest number of balls of radius ϵ required to cover \mathcal{G} . They show that, if $0 < \rho < 1$, then there are classifiers with rate

$$\sup_P \mathbb{E}(d(\hat{G}, G^*)) = O(n^{-1/2}) \quad (38)$$

independent of dimension d . Moreover, if we add the margin (or low noise) assumption

$$\mathbb{P}_X (0 < |m(X) - \frac{1}{2}| \leq t) \leq Ct^\alpha \quad \text{for all } t > 0 \quad (39)$$

we get

$$\sup_P \mathbb{E}(d(\hat{G}, G^*)) = O(n^{-(1+\alpha)/(2+\alpha+\alpha\rho)}) \quad (40)$$

which can be nearly $1/n$ for large α and small ρ . The classifiers can be taken to be plug-in estimators using local polynomial regression. Moreover, they show that this rate is minimax. We will discuss classification in the low noise setting in more detail in another chapter.

9.4 Estimating a Large Covariance Matrix

Let X_1, \dots, X_n be iid Gaussian vectors of dimension d . Let $\Sigma = (\sigma_{ij})_{1 \leq i,j \leq d}$ be the $d \times d$ covariance matrix for X_i . Estimating Σ when d is large is very challenging. Sometimes we can take advantage of special structure. Bickel and Levina (2008) considered the class of *covariance matrices* Σ whose entries have polynomial decay. Specifically, $\Theta = \Theta(\alpha, \epsilon, M)$ is all covariance matrices Σ such that $0 < \epsilon \leq \lambda_{\min}(\Sigma) \leq \lambda_{\max}(\Sigma) \leq 1/\epsilon$ and such that

$$\max_j \sum_i \left\{ |\sigma_{ij}| : |i - j| > k \right\} \leq Mk^{-\alpha}$$

for all k . The loss function is $\|\widehat{\Sigma} - \Sigma\|$ where $\|\cdot\|$ is the operator norm

$$\|A\| = \sup_{x: \|x\|_2=1} \|Ax\|_2.$$

Bickel and Levina (2008) constructed an estimator that converges at rate $(\log d/n)^{\alpha/(\alpha+1)}$. Cai, Zhang and Zhou (2009) showed that the minimax rate is

$$\min \left\{ n^{-2\alpha/(2\alpha+1)} + \frac{\log d}{n}, \frac{d}{n} \right\}$$

so the Bickel-Levina estimator is not rate minimax. Cai, Zhang and Zhou then constructed an estimator that is rate minimax.

9.5 Semisupervised Prediction

Suppose we have data $(X_1, Y_1), \dots, (X_n, Y_n)$ for a classification or regression problem. In addition, suppose we have extra unlabelled data X_{n+1}, \dots, X_N . Methods that make use of the unlabeled are called *semisupervised methods*. We discuss semisupervised methods in another Chapter.

When do the unlabeled data help? Two minimax analyses have been carried out to answer that question, namely, Lafferty and Wasserman (2007) and Singh, Nowak and Zhu (2008). Here we briefly summarize the results of the latter.

Suppose we want to estimate $m(x) = \mathbb{E}(Y|X = x)$ where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. Let p be the density of X . To use the unlabelled data we need to link m and p in some way. A common assumption is the *cluster assumption*: m is smooth over clusters of the marginal $p(x)$. Suppose that p has clusters separated by a amount γ and that m is α smooth over each cluster. Singh, Nowak and Zhu (2008) obtained the following upper and lower minimax bounds as γ varies in 6 zones which we label I to VI. These zones relate the size of γ and the number of unlabeled points:

γ	semisupervised upper bound	supervised lower bound	unlabelled data help?
I	$n^{-2\alpha/(2\alpha+d)}$	$n^{-2\alpha/(2\alpha+d)}$	NO
II	$n^{-2\alpha/(2\alpha+d)}$	$n^{-2\alpha/(2\alpha+d)}$	NO
III	$n^{-2\alpha/(2\alpha+d)}$	$n^{-1/d}$	YES
IV	$n^{-1/d}$	$n^{-1/d}$	NO
V	$n^{-2\alpha/(2\alpha+d)}$	$n^{-1/d}$	YES
VI	$n^{-2\alpha/(2\alpha+d)}$	$n^{-1/d}$	YES

The important message is that there are precise conditions when the unlabeled data help and conditions when the unlabeled data do not help. These conditions arise from computing the minimax bounds.

9.6 Graphical Models

Elsewhere in the book, we discuss the problem of estimating graphical models. Here, we shall briefly mention some minimax results for this problem. Let X be a random vector from a multivariate Normal distribution P with mean vector μ and covariance matrix Σ . Note that X is a random vector of length d , that is, $X = (X_1, \dots, X_d)^T$. The $d \times d$ matrix $\Omega = \Sigma^{-1}$ is called the precision matrix. There is one node for each component of X . The undirected graph associated with P has no edge between X_j and X_j if and only if $\Omega_{jj} = 0$. The edge set is $E = \{(j, k) : \Omega_{jk} \neq 0\}$. The graph is $G = (V, E)$ where $V = \{1, \dots, d\}$ and E is the edge set. Given a random sample of vectors $X^1, \dots, X^n \sim P$ we want to estimate G . (Only the edge set needs to be estimated; the nodes are known.)

Wang, Wainwright and Ramchandran (2010) found the minimax risk for estimating G under zero-one loss. Let $\mathcal{G}_{d,r}(\lambda)$ denote all the multivariate Normals whose graphs have edge sets with degree at most r and such that

$$\min_{(i,j) \in E} \frac{|\Omega_{jk}|}{\sqrt{\Omega_{jj}\Omega_{kk}}} \geq \lambda.$$

The sample complexity $n(d, r, \lambda)$ is the smallest sample size n needed to recover the true graph with high probability. They show that for any $\lambda \in [0, 1/2]$,

$$n(d, r, \lambda) > \max \left\{ \frac{\log \binom{d-r}{2} - 1}{4\lambda^2}, \frac{\log \binom{d}{r} - 1}{\frac{1}{2} \left(\log \left(1 + \frac{r\lambda}{1-\lambda} \right) - \frac{r\lambda}{1+(r-1)\lambda} \right)} \right\}. \quad (41)$$

Thus, assuming $\lambda \approx 1/r$, we get that $n \geq Cr^2 \log(d - r)$.

9.7 Deconvolution and Measurement Error

A problem has seems to have received little attention in the machine learning literature is *deconvolution*. Suppose that $X_1, \dots, X_n \sim P$ where P has density p . We have seen that the minimax rate for estimating p in squared error loss is $n^{-\frac{2\beta}{2\beta+1}}$ where β is the assumed amount of smoothness. Suppose we cannot observe X_i directly but instead we observe X_i with error. Thus, we observe Y_1, \dots, Y_n where

$$Y_i = X_i + \epsilon_i, \quad i = 1, \dots, n. \quad (42)$$

The minimax rates for estimating p change drastically. A good account is given in Fan (1991). As an example, if the noise ϵ_i is Gaussian, then Fan shows that the minimax risk satisfies

$$R_n \geq C \left(\frac{1}{\log n} \right)^\beta$$

which means that the problem is essentially hopeless.

Similar results hold for nonparametric regression. In the usual nonparametric regression problem we observe $Y_i = m(X_i) + \epsilon_i$ and we want to estimate the function m . If we observe $X_i^* = X_i + \delta_i$ instead of X_i then again the minimax rates change drastically and are logarithmic if the δ_i 's are Normal (Fan and Truong 1993). This is known as *measurement error* or *errors in variables*.

This is an interesting example where minimax theory reveals surprising and important insight.

9.8 Normal Means

Perhaps the best understood cases in minimax theory involve normal means. First suppose that $X_1, \dots, X_n \sim N(\theta, \sigma^2)$ where σ^2 is known. A function g is *bowl-shaped* if the sets $\{x : g(x) \leq c\}$ are convex and symmetric about the origin. We will say that a loss function ℓ is bowl-shaped if $\ell(\theta, \hat{\theta}) = g(\theta - \hat{\theta})$ for some bowl-shaped function g .

Theorem 19 *The unique² estimator that is minimax for every bowl-shaped loss function is the sample mean \bar{X}_n .*

For a proof, see Wolfowitz (1950).

Now consider estimating several normal means. Let $X_j = \theta_j + \epsilon_j/\sqrt{n}$ for $j = 1, \dots, n$ and suppose we want to estimate $\theta = (\theta_1, \dots, \theta_n)$ with loss function $\ell(\hat{\theta}, \theta) = \sum_{j=1}^n (\hat{\theta}_j - \theta_j)^2$. Here, $\epsilon_1, \dots, \epsilon_n \sim N(0, \sigma^2)$. This is called the *normal means problem*.

There are strong connections between the normal means problem and nonparametric learning. For example, suppose we want to estimate a regression function $f(x)$ and we observe data $Z_i = f(i/n) + \delta_i$ where $\delta_i \sim N(0, \sigma^2)$. Expand f in an orthonormal basis: $f(x) = \sum_j \theta_j \psi_j(x)$. An estimate of θ_j is $X_j = \frac{1}{n} \sum_{i=1}^n Z_i \psi_j(i/n)$. It follows that $X_j \approx N(\theta_j, \sigma^2/n)$. This connection can be made very rigorous; see Brown and Low (1996).

The minimax risk depends on the assumptions about θ .

Theorem 20 (Pinsker) *1. If $\Theta_n = \mathbb{R}^n$ then $R_n = \sigma^2$ and $\hat{\theta} = X = (X_1, \dots, X_n)$ is minimax.*

2. If $\Theta_n = \{\theta : \sum_j \theta_j^2 \leq C^2\}$ then

$$\liminf_{n \rightarrow \infty} \inf_{\hat{\theta}} \sup_{\theta \in \Theta_n} R(\hat{\theta}, \theta) = \frac{\sigma^2 C^2}{\sigma^2 + C^2}. \quad (43)$$

²Up to sets of measure 0.

Define the James-Stein estimator

$$\widehat{\theta}_{\text{JS}} = \left(1 - \frac{(n-2)\sigma^2}{\frac{1}{n} \sum_{j=1}^n X_j^2} \right) X. \quad (44)$$

Then

$$\lim_{n \rightarrow \infty} \sup_{\theta \in \Theta_n} R(\widehat{\theta}_{\text{JS}}, \theta) = \frac{\sigma^2 C^2}{\sigma^2 + C^2}. \quad (45)$$

Hence, $\widehat{\theta}_{\text{JS}}$ is asymptotically minimax.

3. Let $X_j = \theta_j + \epsilon_j$ for $j = 1, 2, \dots$, where $\epsilon_j \sim N(0, \sigma^2/n)$.

$$\Theta = \left\{ \theta : \sum_{j=1}^{\infty} \theta_j^2 a_j^2 \leq C^2 \right\} \quad (46)$$

where $a_j^2 = (\pi j)^{2p}$. Let R_n denote the minimax risk. Then

$$\min_{n \rightarrow \infty} n^{\frac{2p}{2p+1}} R_n = \left(\frac{\sigma}{\pi} \right)^{\frac{2p}{2p+1}} C^{\frac{2}{2p+1}} \left(\frac{p}{p+1} \right)^{\frac{2p}{2p+1}} (2p+1)^{\frac{1}{2p+1}}. \quad (47)$$

Hence, $R_n \asymp n^{-\frac{2p}{2p+1}}$. An asymptotically minimax estimator is the Pinsker estimator defined by $\widehat{\theta} = (w_1 X_1, w_2 X_2, \dots)$ where $w_j = [1 - (a_j/\mu)]_+$ and μ is determined by the equation

$$\frac{\sigma^2}{n} \sum_j a_j (\mu - a_j)_+ = C^2.$$

The set Θ in (46) is called a *Sobolev ellipsoid*. This set corresponds to smooth functions in the function estimation problem. The Pinsker estimator corresponds to estimating a function by smoothing. The main message to take away from all of this is that minimax estimation under smoothness assumptions requires shrinking the data appropriately.

10 Adaptation

The results in this chapter provide minimax rates of convergence and estimators that achieve these rates. However, the estimators depend on the assumed parameter space. For example, estimating a β -times differential regression function requires using an estimator tailored to the assumed amount of smoothness to achieve the minimax rate $n^{-\frac{2\beta}{2\beta+1}}$. There are estimators that are *adaptive*, meaning that they achieve the minimax rate without the user having to know the amount of smoothness. See, for example, Chapter 9 of Wasserman (2006) and the references therein.

11 Minimax Hypothesis Testing

Let $Y_1, \dots, Y_n \sim P$ where $P \in \mathcal{P}$ and let $P_0 \in \mathcal{P}$. We want to test

$$H_0 : P = P_0 \quad \text{versus} \quad H_1 : P \neq P_0.$$

Recall that a size α test is a function ϕ of (y_1, \dots, y_n) such that $\phi(y_1, \dots, y_n) \in \{0, 1\}$ and $P_0^n(\phi = 1) \leq \alpha$. Let Φ_n be the set level α tests based on n observations where $0 < \alpha < 1$ is fixed. We want to find the minimax type II error

$$\beta_n(\epsilon) = \inf_{\phi \in \Phi_n} \sup_{P \in \mathcal{P}(\epsilon)} P^n(\phi = 0) \quad (48)$$

where

$$\mathcal{P}(\epsilon) = \left\{ P \in \mathcal{P} : d(P_0, Q) > \epsilon \right\}$$

and d is some metric. The *minimax testing rate* is

$$\epsilon_n = \inf \left\{ \epsilon : \beta_n(\epsilon) \leq \delta \right\}.$$

Lower Bound. Define Q by

$$Q(A) = \int P^n(A) d\mu(P) \quad (49)$$

where μ is any distribution whose support is contained in $\mathcal{P}(\epsilon)$. In particular, if μ is uniform on a finite set P_1, \dots, P_N then

$$Q(A) = \frac{1}{N} \sum_j P_j^n(A). \quad (50)$$

Define the likelihood ratio

$$L_n = \frac{dQ}{dP_0^n} = \int \frac{p(y^n)}{p_0(y^n)} d\mu(p) = \int \prod_j \frac{p(y_j)}{p_0(y_j)} d\mu(p). \quad (51)$$

Lemma 21 *Let $0 < \delta < 1 - \alpha$. If*

$$E_0[L_n^2] \leq 1 + 4(1 - \alpha - \delta)^2 \quad (52)$$

then $\beta_n(\epsilon) \geq \delta$.

Proof. Since $P_0^n(\phi = 1) \leq \alpha$ for each ϕ , we have

$$\begin{aligned}\beta_n(\epsilon) &= \inf_{\phi} \sup_{P \in \mathcal{P}(\epsilon)} P^n(\phi = 0) \geq \inf_{\phi} Q(\phi = 0) \geq \inf_{\phi} (P_0^n(\phi = 0) + [Q(\phi = 0) - P_0^n(\phi = 0)]) \\ &\geq 1 - \alpha + [Q(\phi = 0) - P_0^n(\phi = 0)] \geq 1 - \alpha - \sup_A |Q(A) - P_0^n(A)| \\ &= 1 - \alpha - \frac{1}{2} \|Q - P_0^n\|_1.\end{aligned}$$

Now,

$$\|Q - P_0^n\|_1 = \int |L_n(y^n) - 1| dP_0(y^n) = E_0 |L_n(y^n) - 1| \leq \sqrt{E_0[L_n^2] - 1}.$$

The result follows from (52). \square

Upper Bound. Let ϕ be any size α test. Then

$$\beta_n(\epsilon) \leq \sup_{P \in \mathcal{P}(\epsilon)} P^n(\phi = 0)$$

gives an upper bound.

11.1 Multinomials

Let $Y_1, \dots, Y_n \sim P$ where $Y_i \in \{1, \dots, d\} \equiv S$. The minimax estimation rate under L_1 loss for this problem is $O(\sqrt{d/n})$. We will show that the minimax testing rate is $d^{1/4}/\sqrt{n}$.

We will focus on the uniform distribution p_0 . So $p_0(y) = 1/d$ for all y . Let

$$\mathcal{P}(\epsilon) = \left\{ p : \|p - p_0\|_1 > \epsilon \right\}.$$

We want to find ϵ_n such that $\beta_n(\mathcal{P}(\epsilon_n)) = \delta$.

Theorem 22 *Let*

$$\epsilon \leq \frac{Cd^{1/4}}{\sqrt{n}}$$

where

$$C = \frac{1}{2} [\log(1 + 4(1 - \alpha - \delta)^2)]^{1/4}.$$

Assume that $\epsilon \leq 1$. Then $\beta_n(\epsilon) \geq \delta$.

Proof. Let $\gamma = \epsilon/d$. Let $\eta = (\eta_1, \dots, \eta_d)$ where $\eta_j \in \{-1, +1\}$ and $\sum_j \eta_j = 0$. Let R be all such sequences. (This is like a Radamacher sequence except that they are not quite independent.) Define p_η by

$$p_\eta(y) = p_0(y) + \gamma \sum_j \eta_j \psi_j(y)$$

where $\psi_j(y) = 1$ if $y = j$ and $\psi_j(y) = 0$ otherwise. Then $p_\eta(j) \geq 0$ for all j and $\sum_j p_\eta(j) = 1$. Also

$$\|p_0 - p_\eta\|_1 = \sum_j |\gamma \eta_j| = \gamma d = \epsilon.$$

Let N be the number of such probability functions. Now

$$L_n = \frac{1}{N} \sum_{\eta} \prod_i \frac{p_\eta(Y_i)}{p_0(Y_i)}$$

and, for any $\eta, \nu \in R$,

$$\begin{aligned} L_n^2 &= \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \prod_i \frac{p_\eta(Y_i)p_\nu(Y_i)}{p_0(Y_i)p_0(Y_i)} \\ &= \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \prod_i \frac{(p_0(Y_i) + \gamma \sum_j \eta_j \psi_j(Y_i)) (p_0(Y_i) + \gamma \sum_j \nu_j \psi_j(Y_i))}{p_0(Y_i)} \\ &= \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \prod_i \left(1 + d\gamma \sum_j \eta_j \psi_j(Y_i) \right) \left(1 + d\gamma \sum_j \nu_j \psi_j(Y_i) \right). \end{aligned}$$

Taking the expected value over Y_1, \dots, Y_n , and using the fact that $\psi_j(Y_i)\psi_k(Y_i) = 0$ for $j \neq k$, and that $\mathbb{E}_0[\psi_j(Y)] = P_0(Y = j) = 1/d$, we have

$$\begin{aligned} E_0[L_n^2] &= \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \left(1 + d\gamma^2 \sum_j \eta_j \nu_j \right)^n \leq \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \exp(nd\gamma^2 \langle \eta, \nu \rangle) \\ &= E_{\eta, \nu} e^{nd\gamma^2 \langle \eta, \nu \rangle} \end{aligned}$$

where $E_{\eta, \nu}$ denotes expectation over random draws of η and ν . The average over η and ν can be thought of as averages with respect to random assignments of the 1's and -1's. Hence, because these are negatively associated,

$$\begin{aligned} E_0[L_n^2] &\leq E_{\eta, \nu} [e^{nd\gamma^2 \langle \eta, \nu \rangle}] \leq \prod_j E e^{nd\gamma^2 \eta_j \nu_j} \\ &= \prod_j \cosh(nd\gamma^2) \leq \prod_j (1 + 2n^2 d^2 \gamma^4) \leq \prod_j e^{2n^2 d^2 \gamma^4} \\ &= e^{2n^2 d^3 \gamma^4} = e^{2n^2 d^3 (\epsilon^4 / d^4)} \leq 1 + 4(1 - \alpha - \delta)^2 \end{aligned}$$

where we used the fact that, $\gamma = \epsilon_n/d$ and for small x , $\cosh(x) \leq 1 + 2x^2$. From Lemma 21, it follows that $\beta_n(\mathcal{P}(\epsilon, L)) \geq \delta$. \square

For the upper bound we use a test due to Paninski (2008). Let T_n be the number of bins with example one observation. Define t_n by

$$P_0(T_n < t_n) = \alpha.$$

Let $\phi = I(T_n < t_n)$. Thus $\phi \in \Phi_n$.

Lemma 23 *There exists C_1 such that, if $\|p - p_0\| > C_1 d^{1/4}/\sqrt{n}$ then $P(\phi = 0) < \delta$.*

11.2 Testing Densities

Now consider $Y_1, \dots, Y_n \sim P$ where $Y_i \in [0, 1]^d$. Let p_0 be the uniform density. We want to test $H_0 : P = P_0$.

Define

$$\mathcal{P}(\epsilon, s, L) = \left\{ f : \int |f_0 - f| \geq \epsilon \right\} \cap \mathcal{H}_s(L) \quad (53)$$

where $f \in \mathcal{H}_s(L)$ if, for all x, y ,

$$|f^{(t-1)}(y) - f^{(t-1)}(x)| \leq L|x - y|^t$$

and $\|f^{(t-1)}\|_\infty \leq L$ for $t = 1, \dots, s$.

Theorem 24 *Fix $\delta > 0$. Define ϵ_n by $\beta(\epsilon_n) = \delta$. Then, there exist $c_1, c_2 > 0$ such that*

$$c_1 n^{-\frac{2s}{4s+d}} \leq \epsilon_n \leq c_2 n^{-\frac{2s}{4s+d}}.$$

Proof. Here is a proof outline. Divide the space into $k = n^{2/(4s+d)}$ equal size bins. Let η be a Radamacher sequence. Let ψ be a smooth function such that $\int \psi = 0$ and $\int \psi^2 = 1$. For the j^{th} bin B_j , let ψ_j be the function ψ rescaled and recentered to be supported on B_j with $\int \psi_j^2 = 1$. Define

$$p_\eta(y) = p_0(y) + \gamma \sum_j \eta_j \psi_j(y)$$

where $\gamma = cn^{-(2s+d)/(4s+d)}$. It may be verified p_η is a density in $\mathcal{H}_s(L)$ and that $\int |p_0 - p_\eta| \geq \epsilon$. Let N be the number of Rademacher sequences. Then

$$L_n = \frac{1}{N} \sum_{\eta} \prod_i \frac{p_\eta(Y_i)}{p_0(Y_i)}$$

and

$$\begin{aligned}
L_n^2 &= \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \prod_i \frac{p_{\eta}(Y_i)p_{\nu}(Y_i)}{p_0(Y_i)p_0(Y_i)} \\
&= \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \prod_i \frac{(p_0(Y_i) + \sum_j \gamma \eta_j \psi_j(Y_i))}{p_0(Y_i)} \frac{(p_0(Y_i) + \sum_j \gamma \nu_j \psi_j(Y_i))}{p_0(Y_i)} \\
&= \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \prod_i \left(1 + \frac{\sum_j \gamma \eta_j \psi_j(Y_i)}{p_0(Y_i)}\right) \left(1 + \frac{\sum_j \gamma \nu_j \psi_j(Y_i)}{p_0(Y_i)}\right).
\end{aligned}$$

Taking the expected value over Y_1, \dots, Y_n , and using the fact that the ψ_j 's are orthogonal,

$$E_0[L_n^2] = \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \left(1 + \sum_j \gamma^2 \eta_j \nu_j\right)^n \leq \frac{1}{N^2} \sum_{\eta} \sum_{\nu} \exp\left(n \sum_j \gamma^2 \eta_j \nu_j\right).$$

Thus $E_0[L_n^2] \leq E_{\eta, \nu} e^{n\langle \eta, \nu \rangle}$ where $\langle \eta, \nu \rangle = \gamma^2 \sum_j \eta_j \nu_j$. Hence,

$$\begin{aligned}
E_0[L_n^2] &\leq E_{\eta, \nu} e^{n\langle \eta, \nu \rangle} = \prod_j E e^{n\eta_j \nu_j} \\
&= \prod_j \cosh(n\rho_j^2) \leq \prod_j (1 + n^2 \rho_j^4) \leq \prod_j e^{n^2 \gamma^4} = e^{kn^2 \gamma^4} \leq C_0.
\end{aligned}$$

From Lemma 21, it follows that $\beta_n(\mathcal{P}(\epsilon, L)) \geq \delta$. The upper bound can be obtained by using a χ^2 test on the bins. \square

Surprisingly, the story gets much more complicated for non-uniform densities.

12 Summary

Minimax theory allows us to state precisely the best possible performance of any procedure under given conditions. The key tool for finding lower bounds on the minimax risk is Fano's inequality. Finding an upper bound usually involves finding a specific estimator and computing its risk.

13 Bibliographic remarks

There is a vast literature on minimax theory however much of it is scattered in various journal articles. Some texts that contain minimax theory include Tsybakov (2009), van de Geer (2000), van der Vaart (1998) and Wasserman (2006).

References: Arias-Castro Ingster (xxxx), Yu (2008), Tsybakov (2009), van der Vaart (1998), Wasserman (2014).

14 Appendix

14.1 Metrics For Probability Distributions

Minimax theory often makes use of various metrics for probability distributions. Here we summarize some of these metrics and their properties.

Let P and Q be two distributions with densities p and q . We write the distance between P and Q as either $d(P, Q)$ or $d(p, q)$ whichever is convenient. We define the following distances and related quantities.

Total variation	$\text{TV}(P, Q) = \sup_A P(A) - Q(A) $
L_1 distance	$d_1(P, Q) = \int p - q $
L_2 distance	$d_2(P, Q) = \sqrt{\int p - q ^2}$
Hellinger distance	$h(P, Q) = \sqrt{\int (\sqrt{p} - \sqrt{q})^2}$
Kullback-Leibler distance	$\text{KL}(P, Q) = \int p \log(p/q)$
χ^2	$\chi^2(P, Q) = \int (p - q)^2/p$
Affinity	$\ P \wedge Q\ = \int p \wedge q = \int \min\{p(x), q(x)\}dx$
Hellinger affinity	$A(P, Q) = \int \sqrt{pq}$

There are many relationships between these quantities. These are summarized in the next two theorems. We leave the proofs as exercises.

Theorem 25 *The following relationships hold:*

1. $\text{TV}(P, Q) = \frac{1}{2}d_1(P, Q) = 1 - \|p \wedge q\|$. (*Scheff  s Theorem.*)
2. $\text{TV}(P, Q) = P(A) - Q(A)$ where $A = \{x : p(x) > q(x)\}$.
3. $0 \leq h(P, Q) \leq \sqrt{2}$.
4. $h^2(P, Q) = 2(1 - A(P, Q))$.
5. $\|P \wedge Q\| = 1 - \frac{1}{2}d_1(P, Q)$.
6. $\|P \wedge Q\| \geq \frac{1}{2}A^2(P, Q) = \frac{1}{2} \left(1 - \frac{h^2(P, Q)}{2}\right)^2$. (*Le Cam's inequalities.*)
7. $\frac{1}{2}h^2(P, Q) \leq \text{TV}(P, Q) = \frac{1}{2}d_1(P, Q) \leq h(P, Q) \sqrt{1 - \frac{h^2(P, Q)}{4}}$.
8. $\text{TV}(P, Q) \leq \sqrt{\text{KL}(P, Q)/2}$. (*Pinsker's inequality.*)
9. $\int (\log dP/dQ)_+ dP \leq \text{KL}(P, Q) + \sqrt{\text{KL}(P, Q)/2}$.
10. $\|P \wedge Q\| \geq \frac{1}{2}e^{-\text{KL}(P, Q)}$.
11. $\text{TV}(P, Q) \leq h(P, Q) \leq \sqrt{\text{KL}(P, Q)} \leq \sqrt{\chi^2(P, Q)}$.

Let P^n denote the product measure based on n independent samples from P .

Theorem 26 *The following relationships hold:*

1. $h^2(P^n, Q^n) = 2 \left(1 - \left(1 - \frac{h^2(P, Q)}{2} \right)^n \right)$.
2. $\|P^n \wedge Q^n\| \geq \frac{1}{2} A^2(P^n, Q^n) = \frac{1}{2} \left(1 - \frac{1}{2} h^2(P, Q) \right)^{2n}$.
3. $\|P^n \wedge Q^n\| \geq \left(1 - \frac{1}{2} d_1(P, Q) \right)^n$.
4. $\text{KL}(P^n, Q^n) = n \text{KL}(P, Q)$.

14.2 Fano's Lemma

For $0 < p < 1$ define the entropy $h(p) = -p \log p - (1-p) \log(1-p)$ and note that $0 \leq h(p) \leq \log 2$. Let (Y, Z) be a pair of random variables each taking values in $\{1, \dots, N\}$ with joint distribution $P_{Y,Z}$. Then the mutual information is defined to be

$$I(Y; Z) = \text{KL}(P_{Y,Z}, P_Y \times P_Z) = H(Y) - H(Y|Z) \quad (54)$$

where $H(Y) = -\sum_j \mathbb{P}(Y = j) \log \mathbb{P}(Y = j)$ is the entropy of Y and $H(Y|Z)$ is the entropy of Y given Z . We will use the fact that $I(Y; h(Z)) \leq I(Y; Z)$ for an function h .

Lemma 27 *Let Y be a random variable taking values in $\{1, \dots, N\}$. Let $\{P_1, \dots, P_N\}$ be a set of distributions. Let X be drawn from P_j for some $j \in \{1, \dots, N\}$. Thus $P(X \in A|Y = j) = P_j(A)$. Let $Z = g(X)$ be an estimate of Y taking values in $\{1, \dots, N\}$. Then,*

$$H(Y|X) \leq \mathbb{P}(Z \neq Y) \log(N-1) + h(\mathbb{P}(Z = Y)). \quad (55)$$

We follow the proof from Cover and Thomas (1991).

Proof. Let $E = I(Z \neq Y)$. Then

$$H(E, Y|X) = H(Y|X) + H(E|X, Y) = H(Y|X)$$

since $H(E|X, Y) = 0$. Also,

$$H(E, Y|X) = H(E|X) + H(Y|E, X).$$

But $H(E|X) \leq H(E) = h(\mathbb{P}(Z = Y))$. Also,

$$\begin{aligned} H(Y|E, X) &= P(E = 0)H(Y|X, E = 0) + P(E = 1)H(Y|X, E = 1) \\ &\leq P(E = 0) \times 0 + h(\mathbb{P}(Z = Y)) \log(N-1) \end{aligned}$$

since $Y = g(X)$ when $E = 0$ and, when $E = 1$, $H(Y|X, E = 1)$ by the log number of remaining outcomes. Combining these gives $H(Y|X) \leq \mathbb{P}(Z \neq Y) \log(N-1) + h(\mathbb{P}(Z = Y))$. \square

Lemma 28 (*Fano's Inequality*) *Let $\mathcal{P} = \{P_1, \dots, P_N\}$ and $\beta = \max_{j \neq k} \text{KL}(P_j, P_k)$. For any random variable Z taking values on $\{1, \dots, N\}$,*

$$\frac{1}{N} \sum_{j=1}^N P_j(Z \neq j) \geq \left(1 - \frac{n\beta + \log 2}{\log N}\right).$$

Proof. For simplicity, assume that $n = 1$. The general case follows since $\text{KL}(P^n, Q^n) = n\text{KL}(P, Q)$. Let Y have a uniform distribution on $\{1, \dots, N\}$. Given $Y = j$, let X have distribution P_j . This defines a joint distribution P for (X, Y) given by

$$P(X \in A, Y = j) = P(X \in A|Y = j)P(Y = j) = \frac{1}{N}P_j(A).$$

Hence,

$$\frac{1}{N} \sum_{j=1}^N P(Z \neq j|Y = j) = P(Z \neq Y).$$

From (55),

$$\begin{aligned} H(Y|Z) &\leq P(Z \neq Y) \log(N-1) + h(P(Z = Y)) \leq P(Z \neq Y) \log(N-1) + h(1/2) \\ &= P(Z \neq Y) \log(N-1) + \log 2. \end{aligned}$$

Therefore,

$$\begin{aligned} P(Z \neq Y) \log(N-1) &\geq H(Y|Z) - \log 2 = H(Y) - I(Y; Z) - \log 2 \\ &= \log N - I(Y; Z) - \log 2 \geq \log N - \beta - \log 2. \end{aligned} \quad (56)$$

The last inequality follows since

$$I(Y; Z) \leq I(Y; X) = \frac{1}{N} \sum_{j=1}^N \text{KL}(P_j, \bar{P}) \leq \frac{1}{N^2} \sum_{j,k}^N \text{KL}(P_j, P_k) \leq \beta \quad (57)$$

where $\bar{P} = N^{-1} \sum_{j=1}^N P_j$ and we used the convexity of K . Equation (56) shows that

$$P(Z \neq Y) \log(N-1) \geq \log N - \beta - \log 2$$

and the result follows. \square

14.3 Tsybakov's Method

Proof of Theorem 14. Let $X = (X_1, \dots, X_n)$ and let $\psi \equiv \psi(X) \in \{0, 1, \dots, N\}$. Fix $\tau > 0$ and define

$$A_j = \left\{ \frac{dP_0}{dP_j} \geq \tau \right\}.$$

Then

$$\begin{aligned} P_0(\psi \neq 0) &= \sum_{j=1}^N P_0(\psi = j) \geq \sum_{j=1}^N P_0(\psi = j \cap A_j) \\ &= \sum_{j=1}^N \frac{P_0(\psi = j \cap A_j)}{P_j(\psi = j \cap A_j)} P_j(\psi = j \cap A_j) \\ &\geq \tau \sum_{j=1}^N P_j(\psi = j \cap A_j) \\ &\geq \tau \sum_{j=1}^N P_j(\psi = j) - \tau \sum_{j=1}^N P_j(A_j^c) \\ &= \tau N \left(\frac{1}{N} \sum_{j=1}^N P_j(\psi = j) \right) - \tau N \left(\frac{1}{N} \sum_{j=1}^N P_j(A_j^c) \right) \\ &= \tau N(p_0 - a) \end{aligned}$$

where

$$p_0 = \frac{1}{N} \sum_{j=1}^N P_j(\psi = j), \quad a = \frac{1}{N} \sum_{j=1}^N P_j \left(\frac{dP_0}{dP_j} < \tau \right).$$

Hence,

$$\begin{aligned}
\max_{0 \leq j \leq N} P_j(\psi \neq j) &= \max \left\{ P_0(\psi \neq 0), \max_{1 \leq j \leq N} P_j(\psi \neq j) \right\} \\
&\geq \max \left\{ \tau N(p_0 - a), \max_{1 \leq j \leq N} P_j(\psi \neq j) \right\} \\
&\geq \max \left\{ \tau N(p_0 - a), \frac{1}{N} \sum_{j=1}^N P_j(\psi \neq j) \right\} \\
&= \max \left\{ \tau N(p_0 - a), 1 - p_0 \right\} \\
&\geq \min_{0 \leq p \leq 1} \max \left\{ \tau N(p - a), 1 - p \right\} = \frac{\tau N(1 - a)}{1 + \tau N} \\
&= \left(\frac{\tau N}{1 + \tau N} \right) \left[\frac{1}{N} \sum_{j=1}^N P_j \left(\frac{dP_0}{dP_j} \geq \tau \right) \right].
\end{aligned}$$

In Lemma 29 below we show that

$$\frac{1}{N} \sum_{j=1}^N P_j \left(\frac{dP_0}{dP_j} \geq \tau \right) \geq 1 - \frac{a_* + \sqrt{a_*/2}}{\log(1/\tau)}$$

where $a_* = N^{-1} \sum_j K(P_j, P_0)$. Choosing $\tau = 1/\sqrt{N}$ we get

$$\begin{aligned}
\max_{0 \leq j \leq N} P_j(\psi \neq j) &\geq \frac{\sqrt{N}}{1 + \sqrt{N}} \left(1 - \frac{a_* + \sqrt{a_*/2}}{\log(1/\tau)} \right) \\
&= \frac{\sqrt{N}}{1 + \sqrt{N}} \left(1 - \frac{2(a_* + \sqrt{a_*/2})}{\log N} \right) \\
&\geq \frac{1}{2} \left(1 - \frac{1}{4} \right) = \frac{3}{8}.
\end{aligned}$$

By Theorem 3,

$$R_n \geq \frac{s}{2} \frac{3}{8} \geq \frac{s}{8}. \quad \square$$

Lemma 29 *Let $a_* = N^{-1} \sum_j K(P_j, P_0)$. Then,*

$$\frac{1}{N} \sum_{j=1}^N P_j \left(\frac{dP_0}{dP_j} \geq \tau \right) \geq 1 - \frac{a_* + \sqrt{a_*/2}}{\log(1/\tau)}.$$

Proof. Note that

$$\begin{aligned} P_j \left(\frac{dP_0}{dP_j} \geq \tau \right) &= P_j \left(\frac{dP_j}{dP_0} \leq \frac{1}{\tau} \right) \\ &= 1 - P_j \left(\log \frac{dP_j}{dP_0} \geq \log \left(\frac{1}{\tau} \right) \right). \end{aligned}$$

By Markov's inequality,

$$P_j \left(\log \frac{dP_j}{dP_0} \geq \log \left(\frac{1}{\tau} \right) \right) \leq P_j \left(\left[\log \frac{dP_j}{dP_0} \right]_+ \geq \log \left(\frac{1}{\tau} \right) \right) \leq \frac{1}{\log(1/\tau)} \int \left[\log \frac{dP_j}{dP_0} \right]_+ dP_j.$$

According to Pinsker's second inequality (see Thheorem 25 in the appendix and Tsyabakov Lemma 2.5),

$$\int \left[\log \frac{dP_j}{dP_0} \right]_+ dP_j \leq K(P_j, P_0) + \sqrt{K(P_j, K_0)/2}.$$

So

$$P_j \left(\log \frac{dP_j}{dP_0} \geq \log \left(\frac{1}{\tau} \right) \right) \geq 1 - \frac{1}{\log(1/\tau)} \left[K(P_j, P_0) + \sqrt{K(P_j, K_0)/2} \right].$$

Using Jensen's inequality,

$$\frac{1}{N} \sum_j \sqrt{K(P_j, P_0)} \leq \sqrt{\frac{1}{N} \sum_j K(P_j, P_0)} = \sqrt{a_*}.$$

So

$$\begin{aligned} \frac{1}{N} \sum_{j=1}^N P_j \left(\frac{dP_0}{dP_j} \geq \tau \right) &\geq 1 - \frac{1}{\log(1/\tau)} \frac{1}{N} \sum_j K(P_j, P_0) - \frac{1}{\log(1/\tau)} \frac{1}{N} \sum_j \sqrt{K(P_j, P_0)/2} \\ &\geq 1 - \frac{a_*}{\log(1/\tau)} - \frac{\sqrt{a_*/2}}{\log(1/\tau)}. \end{aligned}$$

□

14.4 Assouad's Lemma

Assouad's Lemma is another way to get a lower bound using hypercubes. Let

$$\Omega = \left\{ \omega = (\omega_1, \dots, \omega_N) : \omega_j \in \{0, 1\} \right\}$$

be the set of binary sequences of length N . Let $\mathcal{P} = \{P_\omega : \omega \in \Omega\}$ be a set of 2^N distributions indexed by the elements of Ω . Let $h(\omega, \nu) = \sum_{j=1}^N I(\omega_j \neq \nu_j)$ be the *Hamming distance* between $\omega, \nu \in \Omega$.

Lemma 30 Let $\{P_\omega : \omega \in \Omega\}$ be a set of distributions indexed by ω and let $\theta(P)$ be a parameter. For any $p > 0$ and any metric d ,

$$\max_{\omega \in \Omega} E_\omega \left(d^p(\hat{\theta}, \theta(P_\omega)) \right) \geq \frac{N}{2^{p+1}} \left(\min_{\substack{\omega, \nu \\ h(\omega, \nu) \neq 0}} \frac{d^p(\theta(P_\omega), \theta(P_\nu))}{h(\omega, \nu)} \right) \left(\min_{\substack{\omega, \nu \\ h(\omega, \nu) = 1}} \|P_\omega \wedge P_\nu\| \right). \quad (58)$$

For a proof, see van der Vaart (1998) or Tsybakov (2009).

14.5 The Bayesian Connection

Another way to find the minimax risk and to find a minimax estimator is to use a carefully constructed Bayes estimator. In this section we assume we have a parametric family of densities $\{p(x; \theta) : \theta \in \Theta\}$ and that our goal is to estimate the parameter θ . Since the distributions are indexed by θ , we can write the risk as $R(\theta, \hat{\theta}_n)$ and the maximum risk as $\sup_{\theta \in \Theta} R(\theta, \hat{\theta}_n)$.

Let Q be a prior distribution for θ . The *Bayes risk* (with respect to Q) is defined to be

$$B_Q(\hat{\theta}_n) = \int R(\theta, \hat{\theta}_n) dQ(\theta) \quad (59)$$

and the *Bayes estimator with respect to Q* is the estimator $\bar{\theta}_n$ that minimizes $B_Q(\hat{\theta}_n)$. For simplicity, assume that Q has a density q . The posterior density is then

$$q(\theta | X^n) = \frac{p(X_1, \dots, X_n; \theta) q(\theta)}{m(X_1, \dots, X_n)}$$

where $m(x_1, \dots, x_n) = \int p(x_1, \dots, x_n; \theta) q(\theta) d\theta$.

Lemma 31 The Bayes risk can be written as

$$\int \left(\int L(\theta, \hat{\theta}_n) q(\theta | x_1, \dots, x_n) d\theta \right) m(x_1, \dots, x_n) dx_1 \cdots dx_n.$$

It follows from this lemma that the Bayes estimator can be obtained by finding $\hat{\theta}_n = \hat{\theta}_n(x_1, \dots, x_n)$ to minimize the inner integral $\int L(\theta, \hat{\theta}_n) q(\theta | x_1, \dots, x_n) d\theta$. Often, this is an easy calculation.

Example 32 Suppose that $L(\theta, \hat{\theta}_n) = (\theta - \hat{\theta}_n)^2$. Then the Bayes estimator is the posterior mean $\bar{\theta}_Q = \int \theta q(\theta | x_1, \dots, x_n) d\theta$.

Now we link Bayes estimators to minimax estimators.

Theorem 33 Let $\hat{\theta}_n$ be an estimator. Suppose that (i) the risk function $R(\theta, \hat{\theta}_n)$ is constant as a function of θ and (ii) $\hat{\theta}_n$ is the Bayes estimator for some prior Q . Then $\hat{\theta}_n$ is minimax.

Proof. We will prove this by contradiction. Suppose that $\hat{\theta}_n$ is not minimax. Then there is some other estimator θ' such that

$$\sup_{\theta \in \Theta} R(\theta, \theta') < \sup_{\theta \in \Theta} R(\theta, \hat{\theta}_n). \quad (60)$$

Now,

$$\begin{aligned} B_Q(\theta') &= \int R(\theta, \theta') dQ(\theta) && \text{definition of Bayes risk} \\ &\leq \sup_{\theta \in \Theta} R(\theta, \theta') && \text{average is less than sup} \\ &< \sup_{\theta \in \Theta} R(\theta, \hat{\theta}_n) && \text{from (60)} \\ &= \int R(\theta, \hat{\theta}_n) dQ(\theta) && \text{since risk is constant} \\ &= B_Q(\hat{\theta}_n) && \text{definition of Bayes risk.} \end{aligned}$$

So $B_Q(\theta') < B_Q(\hat{\theta}_n)$. This is a contradiction because $\hat{\theta}_n$ is the Bayes estimator for Q so it must minimize B_Q . \square

Example 34 Let $X \sim \text{Binomial}(n, \theta)$. The mle is X/n . Let $L(\theta, \hat{\theta}_n) = (\theta - \hat{\theta}_n)^2$. Define

$$\hat{\theta}_n = \frac{\frac{X}{n} + \sqrt{\frac{1}{4n}}}{1 + \sqrt{\frac{1}{n}}}.$$

Some calculations show that this is the posterior mean under a $\text{Beta}(\alpha, \beta)$ prior with $\alpha = \beta = \sqrt{n/4}$. By computing the bias and variance of $\hat{\theta}_n$ it can be seen that $R(\theta, \hat{\theta}_n)$ is constant. Since $\hat{\theta}_n$ is Bayes and has constant risk, it is minimax.

Example 35 Let us now show that the sample mean is minimax for the Normal model. Let $X \sim N_p(\theta, I)$ be multivariate Normal with mean vector $\theta = (\theta_1, \dots, \theta_p)$. We will prove that $\hat{\theta}_n = X$ is minimax when $L(\theta, \hat{\theta}_n) = \|\hat{\theta}_n - \theta\|^2$. Assign the prior $Q = N(0, c^2 I)$. Then the posterior is

$$N\left(\frac{c^2 x}{1 + c^2}, \frac{c^2}{1 + c^2} I\right). \quad (61)$$

The Bayes risk $B_Q(\hat{\theta}_n) = \int R(\theta, \hat{\theta}_n) dQ(\theta)$ is minimized by the posterior mean $\tilde{\theta} = c^2 X / (1 + c^2)$. Direct computation shows that $B_Q(\tilde{\theta}) = pc^2 / (1 + c^2)$. Hence, if θ^* is any estimator, then

$$\frac{pc^2}{1 + c^2} = B_Q(\tilde{\theta}) \leq B_Q(\theta^*) = \int R(\theta^*, \theta) dQ(\theta) \leq \sup_{\theta} R(\theta^*, \theta).$$

This shows that $R(\Theta) \geq pc^2 / (1 + c^2)$ for every $c > 0$ and hence

$$R(\Theta) \geq p. \quad (62)$$

But the risk of $\hat{\theta}_n = X$ is p . So, $\hat{\theta}_n = X$ is minimax.

14.6 Nonparametric Maximum Likelihood and the Le Cam Equation

In some cases, the minimax rate can be found by finding ϵ to solve the equation

$$H(\epsilon_n) = n\epsilon_n^2$$

where $H(\epsilon) = \log N(\epsilon)$ and $N(\epsilon)$ is the smallest number of balls of size ϵ in the Hellinger metric needed to cover \mathcal{P} . $H(\epsilon)$ is called the Hellinger entropy of \mathcal{P} . The equation $H(\epsilon) = n\epsilon^2$ is known as the *Le Cam equation*. In this section we consider one case where this is true. For more general versions of this argument, see Shen and Wong (1995), Barron and Yang (1999) and Birgé and Massart (1993).

Our goal is to estimate the density function using maximum likelihood. The loss function is Hellinger distance. Let \mathcal{P} be a set of probability density functions. We have in mind the nonparametric situation where \mathcal{P} does not correspond to some finite dimensional parametric family. Let $N(\epsilon)$ denote the Hellinger covering number of \mathcal{P} . We will make the following assumptions:

(A1) We assume that there exist $0 < c_1 < c_2 < \infty$ such that $c_1 \leq p(x) \leq c_2$ for all x and all $p \in \mathcal{P}$.

(A2) We assume that there exists $a > 0$ such that

$$H(a\epsilon, \mathcal{P}, h) \leq \sup_{p \in \mathcal{P}} H(\epsilon, B(p, 4\epsilon), h)$$

where $B(p, \delta) = \{q : h(p, q) \leq \delta\}$.

(A3) We assume $\sqrt{n}\epsilon_n \rightarrow \infty$ as $n \rightarrow \infty$ where $H(\epsilon_n) \asymp n\epsilon_n^2$.

Assumption (A1) is a very strong and is made only to make the proofs simpler. Assumption (A2) says that the local entropy and global entropy are of the same order. This is typically true in nonparametric models. Assumption (A3) says that the rate of convergence is slower than $O(1/\sqrt{n})$ which again is typical of nonparametric problems. An example of a class \mathcal{P} that satisfies these conditions is

$$\mathcal{P} = \left\{ p : [0, 1] \rightarrow [c_1, c_2] : \int_0^1 p(x)dx = 1, \int_0^1 (p''(x))^2 dx \leq C^2 \right\}.$$

Thanks to (A1) we have,

$$\begin{aligned} \text{KL}(p, q) &\leq \chi^2(p, q) = \int \frac{(p - q)^2}{p} \leq \frac{1}{c_1} \int (p - q)^2 \\ &= \frac{1}{c_1} \int (\sqrt{p} - \sqrt{q})^2 (\sqrt{p} + \sqrt{q})^2 \\ &\leq \frac{4c_2}{c_1} \int (\sqrt{p} - \sqrt{q})^2 = Ch^2(p, q) \end{aligned} \quad (63)$$

where $C = 4c_2/c_1$.

Let ϵ_n solve the Le Cam equation. More precisely, let

$$\epsilon_n = \min \left\{ \epsilon : H \left(\frac{\epsilon}{\sqrt{2C}} \right) \leq \frac{n\epsilon^2}{16C} \right\}. \quad (64)$$

We will show that ϵ_n is the minimax rate.

Upper Bound. To show the upper bound, we will find an estimator that achieves the rate. Let $\mathcal{P}_n = \{p_1, \dots, p_N\}$ be an $\epsilon_n/\sqrt{2C}$ covering set where $N = N(\epsilon_n/\sqrt{2C})$. The set \mathcal{P}_n is an approximation to \mathcal{P} that grows with sample size n . Such a set is called *sieve*. Let \hat{p} be the mle over \mathcal{P}_n , that is, $\hat{p} = \operatorname{argmax}_{p \in \mathcal{P}_n} \mathcal{L}(p)$ where $\mathcal{L}(p) = \prod_{i=1}^n p(X_i)$ is the likelihood function. We call \hat{p} , a *sieve maximum likelihood estimator*. It is crucial that the estimator is computed over \mathcal{P}_n rather than over \mathcal{P} to prevent overfitting. Using a sieve is a type of regularization. We need the following lemma.

Lemma 36 (Wong and Shen) *Let p_0 and p be two densities and let $\delta = h(p_0, p)$. Let Z_1, \dots, Z_n be a sample from p_0 . Then*

$$\mathbb{P} \left(\frac{\mathcal{L}(p)}{\mathcal{L}(p_0)} > e^{-n\delta^2/2} \right) \leq e^{-n\delta^2/4}.$$

Proof.

$$\begin{aligned}
\mathbb{P} \left(\frac{\mathcal{L}(p)}{\mathcal{L}(p_0)} > e^{-n\delta^2/2} \right) &= \mathbb{P} \left(\prod_{i=1}^n \sqrt{\frac{p(Z_i)}{p_0(Z_i)}} > e^{-n\delta^2/4} \right) \leq e^{n\delta^2/4} \mathbb{E} \left(\prod_{i=1}^n \sqrt{\frac{p(Z_i)}{p_0(Z_i)}} \right) \\
&= e^{n\delta^2/4} \left(\mathbb{E} \left(\sqrt{\frac{p(Z_i)}{p_0(Z_i)}} \right) \right)^n = e^{n\delta^2/4} \left(\int \sqrt{p_0(p)} \right)^n \\
&= e^{n\delta^2/4} \left(1 - \frac{h^2(p_0, p)}{2} \right)^n = e^{n\delta^2/4} \exp \left(n \log \left(1 - \frac{h^2(p_0, p)}{2} \right) \right) \\
&\leq e^{n\delta^2/4} e^{-nh^2(p_0, p)/2} = e^{-n\delta^2/4}.
\end{aligned}$$

□

In what follows, we use c, c_1, c_2, \dots , to denote various positive constants.

Theorem 37 $\sup_{P \in \mathcal{P}} \mathbb{E}_p(h(p, \hat{p})) = O(\epsilon_n)$.

Proof. Let p_0 denote the true density. Let p_* be the element of \mathcal{P}_n that minimizes $\text{KL}(p_0, p_j)$. Hence, $\text{KL}(p_0, p_*) \leq Cd^2(p_0, p_*) \leq C(\epsilon_n^2/(2C)) = \epsilon_n^2/2$. Let

$$B = \{p \in \mathcal{P}_n : d(p_*, p) > A\epsilon_n\}$$

where $A = 1/\sqrt{2C}$. Then

$$\begin{aligned}
\mathbb{P}(h(\hat{p}, p_0) > D\epsilon_n) &\leq \mathbb{P}(h(\hat{p}, p_*) + h(p_0, p_*) > D\epsilon_n) \leq \mathbb{P}(h(\hat{p}, p_*) + \frac{\epsilon}{\sqrt{2C}} > D\epsilon_n) \\
&= \mathbb{P}(h(\hat{p}, p_*) > A\epsilon_n) = \mathbb{P}(\hat{p} \in B) \leq \mathbb{P} \left(\sup_{p \in B} \frac{\mathcal{L}(p)}{\mathcal{L}(p_*)} > 1 \right) \\
&\leq \mathbb{P} \left(\sup_{p \in B} \frac{\mathcal{L}(p)}{\mathcal{L}(p_*)} > e^{-n\epsilon_n^2(A^2/2+1)} \right) \\
&\leq \mathbb{P} \left(\sup_{p \in B} \frac{\mathcal{L}(p)}{\mathcal{L}(p_*)} > e^{-n\epsilon_n^2 A^2/2} \right) + \mathbb{P} \left(\sup_{p \in B} \frac{\mathcal{L}(p_0)}{\mathcal{L}(p_*)} > e^{n\epsilon_n^2} \right) \\
&\equiv P_1 + P_2.
\end{aligned}$$

Now

$$P_1 \leq \sum_{p \in B} \mathbb{P} \left(\frac{\mathcal{L}(p)}{\mathcal{L}(p_*)} > e^{-n\epsilon_n^2 A^2/2} \right) \leq N(\epsilon/\sqrt{2C}) e^{-n\epsilon_n^2 A^2/4} \leq e^{n\epsilon_n^2/(16C)}$$

where we used Lemma 36 and the definition of ϵ_n . To bound P_2 , define $K_n = \frac{1}{n} \sum_{i=1}^n \log \frac{p_0(Z_i)}{p_*(Z_i)}$. Hence, $\mathbb{E}(K_n) = \text{KL}(p_0, p_*) \leq \epsilon_n^2/2$. Also,

$$\begin{aligned}
\sigma^2 &\equiv \text{Var} \left(\log \frac{p_0(Z)}{p_*(Z)} \right) \leq \mathbb{E} \left(\log \frac{p_0(Z)}{p_*(Z)} \right)^2 \leq \log \left(\frac{c_2}{c_1} \right) \mathbb{E} \left(\log \frac{p_0(Z)}{p_*(Z)} \right) \\
&= \log \left(\frac{c_2}{c_1} \right) \text{KL}(p_0, p_*) \leq \log \left(\frac{c_2}{c_1} \right) \frac{\epsilon_n^2}{2} \equiv c_3 \epsilon_n^2
\end{aligned}$$

where we used (63). So, by Bernstein's inequality,

$$\begin{aligned}
P_2 &= \mathbb{P}(K_n > \epsilon_n^2) = \mathbb{P}(K_n - \mathsf{KL}(p_0, p_*) > \epsilon_n^2 - \mathsf{KL}(p_0, p_*)) \\
&\leq \mathbb{P}\left(K_n - \mathsf{KL}(p_0, p_*) > \frac{\epsilon_n^2}{2}\right) \leq 2 \exp\left(-\frac{n\epsilon_n^4}{8\sigma^2 + c_4\epsilon_n^2}\right) \\
&\leq 2 \exp(-c_5 n \epsilon_n^2).
\end{aligned}$$

Thus, $P_1 + P_2 \leq \exp(-c_6 n \epsilon_n^2)$. Now,

$$\begin{aligned}
\mathbb{E}(h(\hat{p}, p_0)) &= \int_0^{\sqrt{2}} \mathbb{P}(h(\hat{p}, p_0) > t) dt \\
&= \int_0^{D\epsilon_n} \mathbb{P}(h(\hat{p}, p_0) > t) dt + \int_{D\epsilon_n}^{\sqrt{2}} \mathbb{P}(h(\hat{p}, p_0) > t) dt \\
&\leq D\epsilon_n + \exp(-c_6 n \epsilon_n^2) \leq c_7 \epsilon_n.
\end{aligned}$$

□

Lower Bound. Now we derive the lower bound.

Theorem 38 *Let ϵ_n be the smallest ϵ such that $H(a\epsilon) \geq 64C^2 n \epsilon^2$. Then*

$$\inf_{\hat{p}} \sup_{P \in \mathcal{P}} \mathbb{E}_p(h(p, \hat{p})) = \Omega(\epsilon_n).$$

Proof. Pick any $p \in \mathcal{P}$. Let $B = \{q : h(p, q) \leq 4\epsilon_n\}$. Let $F = \{p_1, \dots, p_N\}$ be an ϵ_n packing set for B . Then

$$N = \log P(\epsilon_n, B, h) \geq \log H(\epsilon_n, B, h) \geq \log H(a\epsilon_n) \geq 64C^2 n \epsilon^2.$$

Hence, for any $P_j, P_k \in F$,

$$\mathsf{KL}(P_j^n, P_k^n) = n \mathsf{KL}(P_j, P_k) \leq C n h^2(P_j, P_k) \leq 16 C n \epsilon_n^2 \leq \frac{N}{4}.$$

It follows from Fano's inequality that

$$\inf_{\hat{p}} \sup_{p \in \mathcal{P}} \mathbb{E}_p h(p, \hat{p}) \geq \frac{1}{4} \min_{j \neq k} h(p_j, p_k) \geq \frac{\epsilon_n}{4}$$

as claimed. □

In summary, we get the minimax rate by solving

$$H(\epsilon_n) \asymp n \epsilon_n^2.$$

Now we can use the Le Cam equation to compute some rates:

Example 39 Here are some standard examples:

Space	Entropy	Rate
Sobolev α	$\epsilon^{-1/\alpha}$	$n^{-\alpha/(2\alpha+1)}$
Sobolev α dimension d	$\epsilon^{-d/\alpha}$	$n^{-\alpha/(2\alpha+d)}$
Lipschitz α	$\epsilon^{-d/\alpha}$	$n^{-\alpha/(2\alpha+d)}$
Monotone	$1/\epsilon$	$n^{-1/3}$
Besov $B_{p,q}^\alpha$	$\epsilon^{-d/\alpha}$	$n^{-\alpha/(2\alpha+d)}$
Neural Nets	see below	see below
m -dimensional parametric model	$m \log(1/\epsilon)$	(m/n)

In the neural net case we have $f(x) = c_0 + \sum_i c_i \sigma(v_i^T x + b_i)$ where $\|c\|_1 \leq C$, $\|v_i\| = 1$ and σ is a step function or a Lipschitz sigmoidal function. Then

$$\left(\frac{1}{\epsilon}\right)^{1/2+1/d} \leq H(\epsilon) \leq \left(\frac{1}{\epsilon}\right)^{1/2+1/(2d)} \log(1/\epsilon) \quad (65)$$

and hence

$$n^{-(1+2/d)/(2+1/d)} (\log n)^{-(1+2/d)(1+1/d)/(2+1/d)} \leq \epsilon_n \leq (n/\log n)^{-(1+1/d)/(2+1/d)}. \quad (66)$$

Nonparametric Bayesian Methods

1 What is Nonparametric Bayes?

In parametric Bayesian inference we have a model $\mathcal{M} = \{f(y|\theta) : \theta \in \Theta\}$ and data $Y_1, \dots, Y_n \sim f(y|\theta)$. We put a prior distribution $\pi(\theta)$ on the parameter θ and compute the posterior distribution using Bayes' rule:

$$\pi(\theta|Y) = \frac{\mathcal{L}_n(\theta)\pi(\theta)}{m(Y)} \quad (1)$$

where $Y = (Y_1, \dots, Y_n)$, $\mathcal{L}_n(\theta) = \prod_i f(Y_i|\theta)$ is the likelihood function and

$$m(y) = m(y_1, \dots, y_n) = \int f(y_1, \dots, y_n|\theta)\pi(\theta)d\theta = \int \prod_{i=1}^n f(y_i|\theta)\pi(\theta)d\theta$$

is the marginal distribution for the data induced by the prior and the model. We call m the induced marginal. The model may be summarized as:

$$\begin{aligned} \theta &\sim \pi \\ Y_1, \dots, Y_n | \theta &\sim f(y|\theta). \end{aligned}$$

We use the posterior to compute a point estimator such as the posterior mean of θ . We can also summarize the posterior by drawing a large sample $\theta_1, \dots, \theta_N$ from the posterior $\pi(\theta|Y)$ and the plotting the samples.

In nonparametric Bayesian inference, we replace the finite dimensional model $\{f(y|\theta) : \theta \in \Theta\}$ with an infinite dimensional model such as

$$\mathcal{F} = \left\{ f : \int (f''(y))^2 dy < \infty \right\} \quad (2)$$

Typically, neither the prior nor the posterior have a density function with respect to a dominating measure. But the posterior is still well defined. On the other hand, if there is a dominating measure for a set of densities \mathcal{F} then the posterior can be found by Bayes theorem:

$$\pi_n(A) \equiv \mathbb{P}(f \in A|Y) = \frac{\int_A \mathcal{L}_n(f)d\pi(f)}{\int_{\mathcal{F}} \mathcal{L}_n(f)d\pi(f)} \quad (3)$$

where $A \subset \mathcal{F}$, $\mathcal{L}_n(f) = \prod_i f(Y_i)$ is the likelihood function and π is a prior on \mathcal{F} . If there is no dominating measure for \mathcal{F} then the posterior still exists but cannot be obtained by simply applying Bayes' theorem. An estimate of f is the posterior mean

$$\hat{f}(y) = \int f(y)d\pi_n(f). \quad (4)$$

A posterior $1 - \alpha$ region is any set A such that $\pi_n(A) = 1 - \alpha$.

Several questions arise:

1. How do we construct a prior π on an infinite dimensional set \mathcal{F} ?
2. How do we compute the posterior? How do we draw random samples from the posterior?
3. What are the properties of the posterior?

The answers to the third question are subtle. In finite dimensional models, the inferences provided by Bayesian methods usually are similar to the inferences provided by frequentist methods. Hence, Bayesian methods inherit many properties of frequentist methods: consistency, optimal rates of convergence, frequency coverage of interval estimates etc. In infinite dimensional models, this is no longer true. The inferences provided by Bayesian methods do not necessarily coincide with frequentist methods and they do not necessarily have properties like consistency, optimal rates of convergence, or coverage guarantees.

2 Distributions on Infinite Dimensional Spaces

To use nonparametric Bayesian inference, we will need to put a prior π on an infinite dimensional space. For example, suppose we observe $X_1, \dots, X_n \sim F$ where F is an unknown distribution. We will put a prior π on the set of all distributions \mathcal{F} . In many cases, we cannot explicitly write down a formula for π as we can in a parametric model. This leads to the following problem: how we we describe a distribution π on an infinite dimensional space? One way to describe such a distribution is to give an explicit algorithm for drawing from the distribution π . In a certain sense, “knowing how to draw from π ” takes the place of “having a formal for π .”

The Bayesian model can be written as

$$\begin{aligned} F &\sim \pi \\ X_1, \dots, X_n | F &\sim F. \end{aligned}$$

The model and the prior induce a marginal distribution m for (X_1, \dots, X_n) ,

$$m(A) = \int \mathbb{P}_F(A) d\pi(F)$$

where

$$\mathbb{P}_F(A) = \int I_A(x_1, \dots, x_n) dF(x_1) \cdots dF(x_n).$$

We call m the induced marginal. Another aspect of describing our Bayesian model will be to give an algorithm for drawing $X = (X_1, \dots, X_n)$ from m .

After we observe the data $X = (X_1, \dots, X_n)$, we are interested in the posterior distribution

$$\pi_n(A) \equiv \pi(F \in A | X_1, \dots, X_n). \quad (5)$$

Once again, we will describe the posterior by giving an algorithm for drawing randomly from it.

To summarize: in some nonparametric Bayesian models, we describe the prior distribution by giving an algorithm for sampling from the prior π , the marginal m and the posterior π_n .

3 Three Nonparametric Problems

We will focus on three specific problems. The four problems and their most common frequentist and Bayesian solutions are:

Statistical Problem	Frequentist Approach	Bayesian Approach
Estimating a cdf	empirical cdf	Dirichlet process
Estimating a density	kernel smoother	Dirichlet process mixture
Estimating a regression function	kernel smoother	Gaussian process

4 Estimating a cdf

Let X_1, \dots, X_n be a sample from an unknown cdf (cumulative distribution function) F where $X_i \in \mathbb{R}$. The usual frequentist estimate of F is the empirical distribution function

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I(X_i \leq x). \quad (6)$$

Recall that for every $\epsilon > 0$ and every F ,

$$\mathbb{P}_F \left(\sup_x |F_n(x) - F(x)| > \epsilon \right) \leq 2e^{-2n\epsilon^2}. \quad (7)$$

Setting $\epsilon_n = \sqrt{\frac{1}{2n} \log \left(\frac{2}{\alpha} \right)}$ we have

$$\inf_F \mathbb{P}_F \left(F_n(x) - \epsilon_n \leq F(x) \leq F_n(x) + \epsilon_n \text{ for all } x \right) \geq 1 - \alpha \quad (8)$$

where the infimum is over all cdf's F . Thus, $(F_n(x) - \epsilon_n, F_n(x) + \epsilon_n)$ is a $1 - \alpha$ confidence band for F .

To estimate F from a Bayesian perspective we put a prior π on the set of all cdf's \mathcal{F} and then we compute the posterior distribution on \mathcal{F} given $X = (X_1, \dots, X_n)$. The most commonly used prior is the Dirichlet process prior which was invented by the statistician Thomas Ferguson in 1973.

The distribution π has two parameters, F_0 and α and is denoted by $\text{DP}(\alpha, F_0)$. The parameter F_0 is a distribution function and should be thought of as a prior guess at F . The number α controls how tightly concentrated the prior is around F_0 . The model may be summarized as:

$$\begin{aligned} F &\sim \pi \\ X_1, \dots, X_n | F &\sim F \end{aligned}$$

where $\pi = \text{DP}(\alpha, F_0)$.

How to Draw From the Prior. To draw a single random distribution F from $\text{Dir}(\alpha, F_0)$ we do the following steps:

1. Draw s_1, s_2, \dots independently from F_0 .
2. Draw $V_1, V_2, \dots \sim \text{Beta}(1, \alpha)$.
3. Let $w_1 = V_1$ and $w_j = V_j \prod_{i=1}^{j-1} (1 - V_i)$ for $j = 2, 3, \dots$
4. Let F be the discrete distribution that puts mass w_j at s_j , that is, $F = \sum_{j=1}^{\infty} w_j \delta_{s_j}$ where δ_{s_j} is a point mass at s_j .

It is clear from this description that F is discrete with probability one. The construction of the weights w_1, w_2, \dots is often called the stick breaking process. Imagine we have a stick of unit length. Then w_1 is obtained by breaking the stick at the random point V_1 . The stick now has length $1 - V_1$. The second weight w_2 is obtained by breaking a proportion V_2 from the remaining stick. The process continues and generates the whole sequence of weights w_1, w_2, \dots . See Figure 1. It can be shown that if $F \sim \text{Dir}(\alpha, F_0)$ then the mean is $\mathbb{E}(F) = F_0$.

You might wonder why this distribution is called a Dirichlet process. The reason is this. Recall that a random vector $P = (P_1, \dots, P_k)$ has a Dirichlet distribution with parameters $(\alpha, g_1, \dots, g_k)$ (with $\sum_j g_j = 1$) if the distribution of P has density

$$f(p_1, \dots, p_k) = \frac{\Gamma(\alpha)}{\prod_{j=1}^k \Gamma(\alpha g_j)} \prod_{j=1}^k p_j^{\alpha g_j - 1}$$

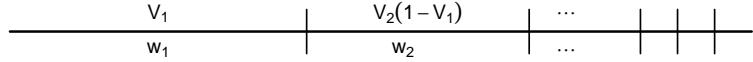


Figure 1: The stick breaking process shows how the weights w_1, w_2, \dots from the Dirichlet process are constructed. First we draw $V_1, V_2, \dots \sim \text{Beta}(1, \alpha)$. Then we set $w_1 = V_1$, $w_2 = V_2(1 - V_1)$, $w_3 = V_3(1 - V_1)(1 - V_2), \dots$

over the simplex $\{p = (p_1, \dots, p_k) : p_j \geq 0, \sum_j p_j = 1\}$. Let (A_1, \dots, A_k) be any partition of \mathbb{R} and let $F \sim \text{DP}(\alpha, F_0)$ be a random draw from the Dirichlet process. Let $F(A_j)$ be the amount of mass that F puts on the set A_j . Then $(F(A_1), \dots, F(A_k))$ has a Dirichlet distribution with parameters $(\alpha, F_0(A_1), \dots, F_0(A_k))$. In fact, this property characterizes the Dirichlet process.

How to Sample From the Marginal. One way is to draw from the induced marginal m is to sample $F \sim \pi$ (as described above) and then draw X_1, \dots, X_n from F . But there is an alternative method, called the Chinese Restaurant Process or infinite Pólya urn (Blackwell 1973). The algorithm is as follows.

1. Draw $X_1 \sim F_0$.
2. For $i = 2, \dots, n$: draw

$$X_i | X_1, \dots, X_{i-1} = \begin{cases} X \sim F_{i-1} & \text{with probability } \frac{i-1}{i+\alpha-1} \\ X \sim F_0 & \text{with probability } \frac{\alpha}{i+\alpha-1} \end{cases}$$

where F_{i-1} is the empirical distribution of X_1, \dots, X_{i-1} .

The sample X_1, \dots, X_n is likely to have ties since F is discrete. Let X_1^*, X_2^*, \dots denote the unique values of X_1, \dots, X_n . Define cluster assignment variables c_1, \dots, c_n where $c_i = j$ means that X_i takes the value X_j^* . Let $n_j = |\{i : c_i = j\}|$. Then we can write

$$X_n = \begin{cases} X_j^* & \text{with probability } \frac{n_j}{n+\alpha-1} \\ X \sim F_0 & \text{with probability } \frac{\alpha}{n+\alpha-1} \end{cases}$$

In the metaphor of the Chinese restaurant process, when the n th customer walks into the restaurant, he sits at table j with probability $n_j/(n + \alpha - 1)$, and occupies a new table with probability $\alpha/(n + \alpha - 1)$. The j th table is associated with a “dish” $X_j^* \sim F_0$. Since the process is exchangeable, it induces (by ignoring X_j^*) a partition over the integers $\{1, \dots, n\}$, which corresponds to a clustering of the indices. See Figure 2.

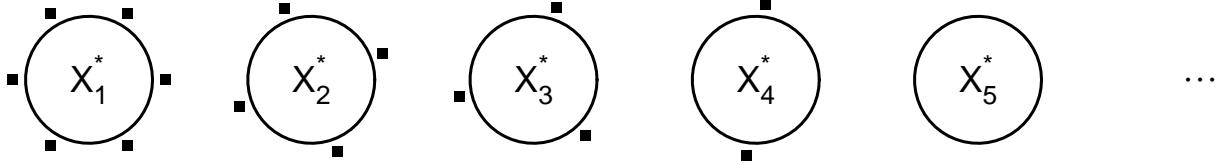


Figure 2: The Chinese restaurant process. A new person arrives and either sits at a table with people or sits at a new table. The probability of sitting at a table is proportional to the number of people at the table.

How to Sample From the Posterior. Now suppose that $X_1, \dots, X_n \sim F$ and that we place a $\text{Dir}(\alpha, F_0)$ prior on F .

Theorem 1 Let $X_1, \dots, X_n \sim F$ and let F have prior $\pi = \text{Dir}(\alpha, F_0)$. Then the posterior π for F given X_1, \dots, X_n is $\text{Dir}(\alpha + n, \bar{F}_n)$ where

$$\bar{F}_n = \frac{n}{n + \alpha} F_n + \frac{\alpha}{n + \alpha} F_0. \quad (9)$$

Since the posterior is again a Dirichlet process, we can sample from it as we did the prior but we replace α with $\alpha + n$ and we replace F_0 with \bar{F}_n . Thus the posterior mean is \bar{F}_n is a convex combination of the empirical distribution and the prior guess F_0 . Also, the predictive distribution for a new observation X_{n+1} is given by \bar{F}_n .

To explore the posterior distribution, we could draw many random distribution functions from the posterior. We could then numerically construct two functions L_n and U_n such that

$$\pi(L_n(x) \leq F(x) \leq U_n(x) \text{ for all } x | X_1, \dots, X_n) = 1 - \alpha.$$

This is a $1 - \alpha$ Bayesian confidence band for F . Keep in mind that this is not a frequentist confidence band. It does *not* guarantee that

$$\inf_F \mathbb{P}_F(L_n(x) \leq F(x) \leq U_n(x) \text{ for all } x) = 1 - \alpha.$$

When n is large, $\bar{F}_n \approx F_n$ in which case there is little difference between the Bayesian and frequentist approach. The advantage of the frequentist approach is that it does not require specifying α or F_0 .

Example 2 Figure 3 shows a simple example. The prior is $\text{DP}(\alpha, F_0)$ with $\alpha = 10$ and $F_0 = N(0, 1)$. The top left plot shows the discrete probability function resulting from a single

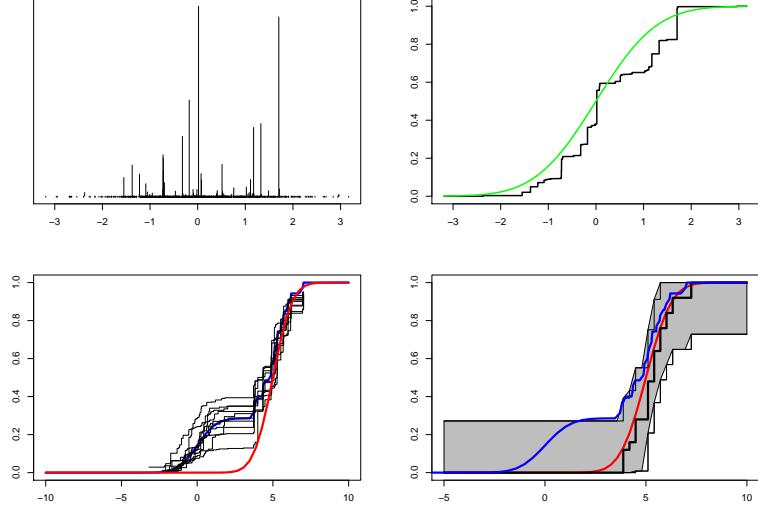


Figure 3: The top left plot shows the discrete probability function resulting from a single draw from the prior which is a $\text{DP}(\alpha, F_0)$ with $\alpha = 10$ and $F_0 = N(0, 1)$. The top right plot shows the resulting cdf along with F_0 . The bottom left plot shows a few draws from the posterior based on $n = 25$ observations from a $N(5, 1)$ distribution. The blue line is the posterior mean and the red line is the true F . The posterior is biased because of the prior. The bottom right plot shows the empirical distribution function (solid black) the true F (red) the Bayesian posterior mean (blue) and a 95 percent frequentist confidence band.

draw from the prior. The top right plot shows the resulting cdf along with F_0 . The bottom left plot shows a few draws from the posterior based on $n = 25$ observations from a $N(5, 1)$ distribution. The blue line is the posterior mean and the red line is the true F . The posterior is biased because of the prior. The bottom right plot shows the empirical distribution function (solid black) the true F (red) the Bayesian posterior mean (blue) and a 95 percent frequentist confidence band.

5 Density Estimation

Let $X_1, \dots, X_n \sim F$ where F has density f and $X_i \in \mathbb{R}$. Our goal is to estimate f . The Dirichlet process is not a useful prior for this problem since it produces discrete distributions which do not even have densities. Instead, we use a modification of the Dirichlet process. But first, let us review the frequentist approach.

The most common frequentist estimator is the kernel estimator

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x - X_i}{h}\right)$$

where K is a kernel and h is the bandwidth. A related method for estimating a density is to use a mixture model

$$f(x) = \sum_{j=1}^k w_j f(x; \theta_j).$$

For example, if $f(x; \theta)$ is Normal then $\theta = (\mu, \sigma)$. The kernel estimator can be thought of as a mixture with n components. In the Bayesian approach we would put a prior on $\theta_1, \dots, \theta_k$, on w_1, \dots, w_k and a prior on k . We could be more ambitious and use an infinite mixture

$$f(x) = \sum_{j=1}^{\infty} w_j f(x; \theta_j).$$

As a prior for the parameters we could take $\theta_1, \theta_2, \dots$ to be drawn from some F_0 and we could take w_1, w_2, \dots , to be drawn from the stick breaking prior. (F_0 typically has parameters that require further priors.) This infinite mixture model is known as the Dirichlet process mixture model. This infinite mixture is the same as the random distribution $F \sim \text{DP}(\alpha, F_0)$ which had the form $F = \sum_{j=1}^{\infty} w_j \delta_{\theta_j}$ except that the point mass distributions δ_{θ_j} are replaced by smooth densities $f(x|\theta_j)$.

The model may be re-expressed as:

$$F \sim \text{DP}(\alpha, F_0) \tag{10}$$

$$\theta_1, \dots, \theta_n | F \sim F \tag{11}$$

$$X_i | \theta_i \sim f(x|\theta_i), \quad i = 1, \dots, n. \tag{12}$$

(In practice, F_0 itself has free parameters which also require priors.) Note that in the DPM, *the parameters θ_i of the mixture are sampled from a Dirichlet process. The data X_i are not sampled from a Dirichlet process.* Because F is sampled from a Dirichlet process, it will be discrete. Hence there will be ties among the θ_i 's. (Recall our earlier discussion of the Chinese Restaurant Process.) The $k < n$ distinct values of θ_i can be thought of as defining clusters. The beauty of this model is that the discreteness of F automatically creates a clustering of the θ_j 's. In other words, we have implicitly created a prior on k , the number of distinct θ_j 's.

How to Sample From the Prior. Draw $\theta_1, \theta_2, \dots, F_0$ and draw w_1, w_2, \dots from the stick breaking process. Set $f(x) = \sum_{j=1}^{\infty} w_j f(x|\theta_j)$. The density f is a random draw from the prior. We could repeat this process many times resulting in many randomly drawn densities from the prior. Plotting these densities could give some intuition about the structure of the prior.

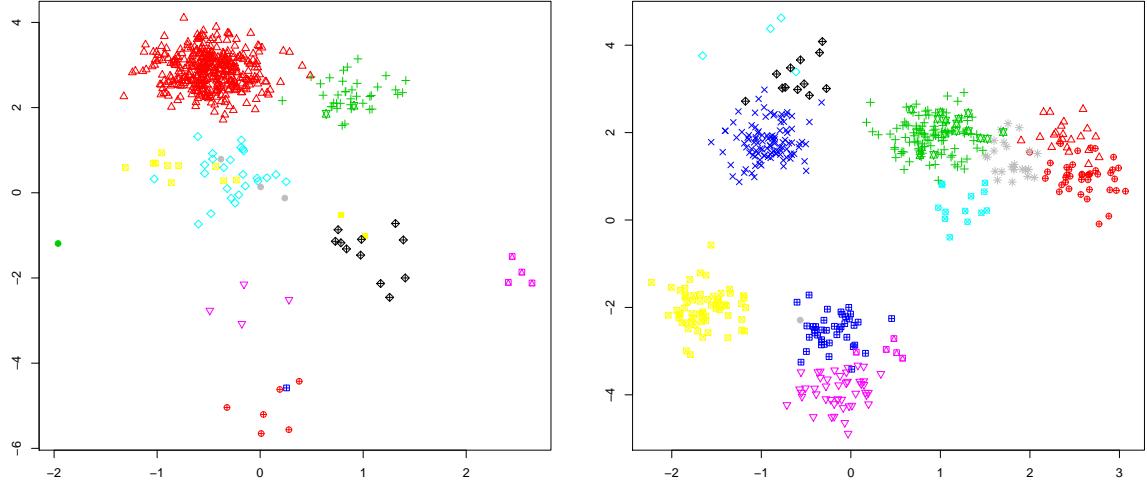


Figure 4: Samples from a Dirichlet process mixture model with Gaussian generator, $n = 500$.

How to Sample From the Prior Marginal. The prior marginal m is

$$m(x_1, x_2, \dots, x_n) = \int \prod_{i=1}^n f(x_i|F) d\pi(F) \quad (13)$$

$$= \int \prod_{i=1}^n \left(\int f(x_i|\theta) p(\theta|F) dF(\theta) \right) dP(G) \quad (14)$$

If we want to draw a sample from m , we first draw F from a Dirichlet process with parameters α and F_0 , and then generate θ_i independently from this realization. Then we sample $X_i \sim f(x|\theta_i)$.

As before, we can also use the Chinese restaurant representation to draw the θ_j 's sequentially. Given $\theta_1, \dots, \theta_{i-1}$ we draw θ_j from

$$\alpha F_0(\cdot) + \sum_{i=1}^{n-1} \delta_{\theta_i}(\cdot). \quad (15)$$

Let θ_j^* denote the unique values among the θ_i , with n_j denoting the number of elements in the cluster for parameter θ_j^* ; that is, if c_1, c_2, \dots, c_{n-1} denote the cluster assignments $\theta_i = \theta_{c_i}^*$ then $n_j = |\{i : c_i = j\}|$. Then we can write

$$\theta_n = \begin{cases} \theta_j^* & \text{with probability } \frac{n_j}{n+\alpha-1} \\ \theta \sim F_0 & \text{with probability } \frac{\alpha}{n+\alpha-1}. \end{cases} \quad (16)$$

How to Sample From the Posterior. We sample from the posterior by Gibbs sampling; we may discuss that later.

To understand better how to use the model, we consider how to use the DPM for estimating density using a mixture of Normals. There are numerous implementations. We consider one due to Ishwaran et al. (2002). The first step (in this particular approach) is to replace the infinite mixture with a large but finite mixture. Thus we replace the stick-breaking process with $V_1, \dots, V_{N-1} \sim \text{Beta}(1, \alpha)$ and $w_1 = V_1, w_2 = V_2(1 - V_1), \dots$. This generates w_1, \dots, w_N which sum to 1. Replacing the infinite mixture with the finite mixture is a numerical trick not an inferential step and has little numerical effect as long as N is large. For example, they show that when $n = 1,000$ it suffices to use $N = 50$. A full specification of the resulting model, including priors on the hyperparameters is:

$$\begin{aligned}\theta &\sim N(0, A) \\ \alpha &\sim \text{Gamma}(\eta_1, \eta_2) \\ \mu_1, \dots, \mu_N &\sim N(\theta, B^2) \\ \frac{1}{\sigma_1^2}, \dots, \frac{1}{\sigma_N^2} &\sim \text{Gamma}(\nu_1, \nu_2) \\ K_1, \dots, K_n &\sim \sum_{j=1}^N w_j \delta_j \\ X_i &\sim N(\mu_i, \sigma_i^2) \quad i = 1, \dots, n\end{aligned}$$

The hyperparameters $A, B, \gamma_1, \gamma_2, \nu_1, \nu_2$ still need to be set. Compare this to kernel density estimation which requires the single bandwidth h . Ishwaran et al use $A = 1000$, $\nu_1 = \nu_2 = \eta_1 = \eta_2 = 2$ and they take B to be 4 times the standard deviation of the data. It is now possible to write down a Gibbs sampling algorithm for sampling from the posterior.

6 Nonparametric Regression

Consider the nonparametric regression model

$$Y_i = m(X_i) + \epsilon_i, \quad i = 1, \dots, n \tag{17}$$

where $\mathbb{E}(\epsilon_i) = 0$. The frequentist kernel estimator for m is

$$\hat{m}(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{\|x-X_i\|}{h}\right)}{\sum_{i=1}^n K\left(\frac{\|x-X_i\|}{h}\right)} \tag{18}$$

where K is a kernel and h is a bandwidth. The Bayesian version requires a prior π on the set of regression functions \mathcal{M} . A common choice is the Gaussian process prior.

A stochastic process $m(x)$ indexed by $x \in \mathcal{X} \subset \mathbb{R}^d$ is a *Gaussian process* if for each $x_1, \dots, x_n \in \mathcal{X}$ the vector $(m(x_1), m(x_2), \dots, m(x_n))$ is Normally distributed:

$$(m(x_1), m(x_2), \dots, m(x_n)) \sim N(\mu(x), K(x)) \tag{19}$$

where $K_{ij}(x) = K(x_i, x_j)$ is a Mercer kernel.

Let's assume that $\mu = 0$. Then for given x_1, x_2, \dots, x_n the density of the Gaussian process prior of $m = (m(x_1), \dots, m(x_n))$ is

$$\pi(m) = (2\pi)^{-n/2} |K|^{-1/2} \exp\left(-\frac{1}{2} m^T K^{-1} m\right) \quad (20)$$

Under the change of variables $m = K\alpha$, we have that $\alpha \sim N(0, K^{-1})$ and thus

$$\pi(\alpha) = (2\pi)^{-n/2} |K|^{-1/2} \exp\left(-\frac{1}{2} \alpha^T K \alpha\right) \quad (21)$$

Under the additive Gaussian noise model, we observe $Y_i = m(x_i) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$. Thus, the log-likelihood is

$$\log p(y|m) = -\frac{1}{2\sigma^2} \sum_i (y_i - m(x_i))^2 + \text{const} \quad (22)$$

and the log-posterior is

$$\log p(y|m) + \log \pi(m) = -\frac{1}{2\sigma^2} \|y - K\alpha\|_2^2 - \frac{1}{2} \alpha^T K \alpha + \text{const} \quad (23)$$

$$= -\frac{1}{2\sigma^2} \|y - K\alpha\|_2^2 - \frac{1}{2} \|\alpha\|_K^2 + \text{const} \quad (24)$$

What functions have high probability according to the Gaussian process prior? The prior favors $\alpha^T K^{-1} \alpha$ being small. Suppose we consider an eigenvector v of K , with eigenvalue λ , so that $Kv = \lambda v$. Then we have that

$$\frac{1}{\lambda} = v^T K^{-1} v \quad (25)$$

Thus, eigenfunctions with *large* eigenvalues are favored by the prior. These correspond to smooth functions; the eigenfunctions that are very wiggly correspond to small eigenvalues.

In this Bayesian setup, MAP estimation corresponds to Mercer kernel regression, which regularizes the squared error by the RKHS norm $\|\alpha\|_K^2$. The posterior mean is

$$\mathbb{E}(\alpha|Y) = (K + \sigma^2 I)^{-1} Y \quad (26)$$

and thus

$$\hat{m} = \mathbb{E}(m|Y) = K (K + \sigma^2 I)^{-1} Y. \quad (27)$$

We see that \hat{m} is nothing but a linear smoother and is, in fact, very similar to the frequentist kernel smoother.

Unlike kernel regression, where we just need to choose a bandwidth h , here we need to choose the function $K(x, y)$. This is a delicate matter.

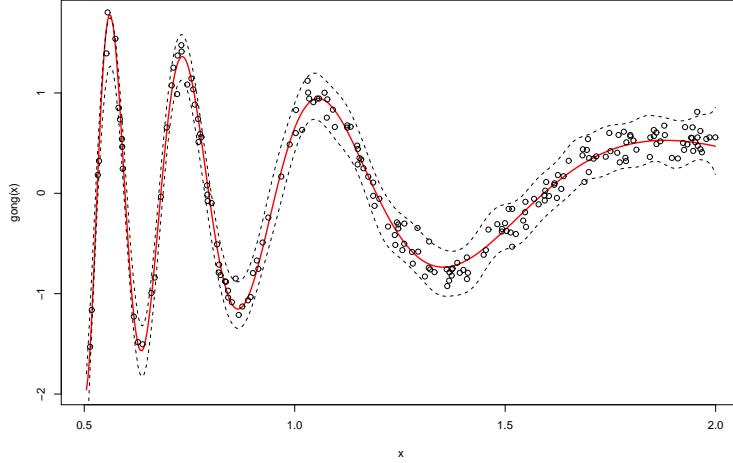


Figure 5: Mean of a Gaussian process

Now, to compute the predictive distribution for a new point $Y_{n+1} = m(x_{n+1}) + \epsilon_{n+1}$, we note that $(Y_1, \dots, Y_n) \sim N(0, (K + \sigma^2 I)\alpha)$. Let k be the vector

$$k = (K(x_1, x_{n+1}), \dots, K(x_n, x_{n+1})) \quad (28)$$

Then (Y_1, \dots, Y_{n+1}) is jointly Gaussian with covariance

$$\begin{pmatrix} K + \sigma^2 I & k \\ k^T & k(x_{n+1}, x_{n+1}) + \sigma^2 \end{pmatrix} \quad (29)$$

Therefore, conditional distribution of Y_{n+1} is

$$Y_{n+1}|Y_{1:n}, x_{1:n} \sim N(k^T(K + \sigma^2 I)^{-1}Y, k(x_{n+1}, x_{n+1}) + \sigma^2 - k^T(K + \sigma^2 I)^{-1}k) \quad (30)$$

Note that the above variance differs from the variance estimated using the frequentist method. However, Bayesian Gaussian process regression and kernel regression often lead to similar results. The advantages of the kernel regression is that it requires a single parameter h that can be chosen by cross-validation and its theoretical properties are simple and well-understood.

7 Theoretical Properties of Nonparametric Bayes

In this section we briefly discuss some theoretical properties of nonparametric Bayesian methods. We will focus on density estimation. In frequentist nonparametric inference, procedures are required to have certain guarantees such as consistency and minimaxity. Similar reasoning can be applied to Bayesian procedures. It is desirable, for example, that

the posterior distribution π_n has mass that is concentrated near the true density function f . More specifically, we can ask three specific questions:

1. Is the posterior consistent?
2. Does posterior concentrate at the optimal rate?
3. Does posterior have correct coverage?

7.1 Consistency

Let f_0 denote the true density. By consistency we mean that, when $f_0 \in A$, $\pi_n(A)$ should converge, in some sense, to 1. According to Doob's theorem, consistency holds under very weak conditions.

To state Doob's theorem we need some notation. The prior π and the model define a joint distribution μ_n on sequences $Y^n = (Y_1, \dots, Y_n)$, namely, for any $B \in \mathbb{R}^n$,¹

$$\mu_n(Y_n \in B) = \int \mathbb{P}(Y^n \in B | f) d\pi(f) = \int_B f(y_1) \cdots f(y_n) d\pi(f). \quad (31)$$

In fact, the model and prior determine a joint distribution μ on the set of infinite sequences² $\mathcal{Y}^\infty = \{Y^\infty = (y_1, y_2, \dots,)\}$.

Theorem 3 (Doob 1949) *For every measurable A ,*

$$\mu \left(\lim_{n \rightarrow \infty} \pi_n(A) = I(f_0 \in A) \right) = 1. \quad (32)$$

By Doob's theorem, consistency holds except on a set of probability zero. This sounds good but it isn't; consider the following example.

Example 4 *Let $Y_1, \dots, Y_n \sim N(\theta, 1)$. Let the prior π be a point mass at $\theta = 0$. Then the posterior is point mass at $\theta = 0$. This posterior is inconsistent on the set $N = \mathbb{R} - \{0\}$. This set has probability 0 under the prior so this does not contradict Doob's theorem. But clearly the posterior is useless.*

Doob's theorem is useless for our purposes because it is solipsistic. The result is with respect to the Bayesian's own distribution μ . Instead, we want to say that the posterior is consistent with respect to \mathbb{P}_0 , the distribution generating the data.

¹More precisely, for any Borel set B .

²More precisely, on an appropriate σ -field over the set of infinite sequences.

To continue, let us define three types of neighborhoods. Let f be a density and let P_f be the corresponding probability measure. A Kullback-Leibler neighborhood around P_f is

$$B_K(p, \epsilon) = \left\{ P_g : \int f(x) \log \left(\frac{f(x)}{g(x)} \right) dx \leq \epsilon \right\}. \quad (33)$$

A Hellinger neighborhood around P_f is

$$B_H(p, \epsilon) = \left\{ P_g : \int (\sqrt{f}(x) - \sqrt{g}(x))^2 \leq \epsilon^2 \right\}. \quad (34)$$

A weak neighborhood around P_f is

$$B_W(P, \epsilon) = \left\{ Q : d_W(P, Q) \leq \epsilon \right\} \quad (35)$$

where d_W is the Prohorov metric

$$d_W(P, Q) = \inf \left\{ \epsilon > 0 : P(B) \leq Q(B^\epsilon) + \epsilon, \text{ for all } B \right\} \quad (36)$$

where $B^\epsilon = \{x : \inf_{y \in B} \|x - y\| \leq \epsilon\}$. Weak neighborhoods are indeed very weak: if $P_g \in B_W(P_f, \epsilon)$ it does not imply that g resembles f .

Theorem 5 (Schwartz 1963) *If*

$$\pi(B_K(f_0, \epsilon)) > 0, \quad \text{for all } \epsilon > 0 \quad (37)$$

then, for any $\delta > 0$,

$$\pi_n(B_W(P, \delta)) \xrightarrow{\text{a.s.}} 1 \quad (38)$$

with respect to P_0 .

This is still unsatisfactory since weak neighborhoods are large. Let $N(\mathcal{M}, \epsilon)$ denote the smallest number of functions f_1, \dots, f_N such that, for each $f \in \mathcal{M}$, there is a f_j such that $f(x) \leq f_j(x)$ for all x and such that $\sup_x (f_j(x) - f(x)) \leq \epsilon$. Let $H(\mathcal{M}, \epsilon) = \log N(\mathcal{M}, \epsilon)$.

Theorem 6 (Barron, Schervish and Wasserman (1999) and Ghosal, Ghosh and Ramamoorthi) *Suppose that*

$$\pi(B_K(f_0, \epsilon)) > 0, \quad \text{for all } \epsilon > 0. \quad (39)$$

Further, suppose there exists $\mathcal{M}_1, \mathcal{M}_2, \dots$ such that $\pi(\mathcal{M}_j^c) \leq c_1 e^{-jc_2}$ and $H(\mathcal{M}_j, \delta) \leq c_3 j$ for all large j . Then, for any $\delta > 0$,

$$\pi_n(B_H(P, \delta)) \xrightarrow{\text{a.s.}} 1 \quad (40)$$

with respect to P_0 .

Example 7 Recall the Normal means model

$$Y_i = \theta_i + \frac{1}{\sqrt{n}}\epsilon_i, \quad i = 1, 2, \dots \quad (41)$$

where $\epsilon_i \sim N(0, \sigma^2)$. We want to infer $\theta = (\theta_1, \theta_2, \dots)$. Assume that θ is contained in the Sobolev space

$$\theta \in \Theta = \left\{ \theta : \sum_i \theta_i^2 i^{2p} < \infty \right\}. \quad (42)$$

Recall that the estimator $\hat{\theta}_i = b_i Y_i$ is minimax for this Sobolev space where b_i is an appropriate constant. In fact the Efromovich-Pinsker estimator is adaptive minimax over the smoothness index p . A simple Bayesian analysis is to use the prior π that treats each θ_i as independent random variables and $\theta_i \sim N(0, \tau_i^2)$ where $\tau_i^2 = i^{-2q}$. Have we really defined a prior on Θ ? We need to make sure that $\pi(\Theta) = 1$. Fix $K > 0$. Then,

$$\pi\left(\sum_i \theta_i^2 i^{2p} > K\right) \leq \frac{\sum_i \mathbb{E}_\pi(\theta_i^2) i^{2p}}{K} = \frac{\sum_i \tau_i^2 i^{2p}}{K} = \frac{\sum_i \frac{1}{i^{2(q-p)}}}{K}. \quad (43)$$

The numerator is finite as long as $q > p + (1/2)$. Assuming $q > p + (1/2)$ we then see that $\pi(\sum_i \theta_i^2 i^{2p} > K) \rightarrow 0$ as $K \rightarrow \infty$ which shows that π puts all its mass on Θ . But, as we shall see below, the condition $q > p + (1/2)$ is guaranteed to yield a posterior with a suboptimal rate of convergence.

7.2 Rates of Convergence

Here the situation is more complicated. Recall the Normal means model

$$Y_i = \theta_i + \frac{1}{\sqrt{n}}\epsilon_i, \quad i = 1, 2, \dots \quad (44)$$

where $\epsilon_i \sim N(0, \sigma^2)$. We want to infer $\theta = (\theta_1, \theta_2, \dots) \in \Theta$ from $Y = (Y_1, Y_2, \dots)$. Assume that θ is contained in the Sobolev space

$$\theta \in \Theta = \left\{ \theta : \sum_i \theta_i^2 i^{2p} < \infty \right\}. \quad (45)$$

The following results are from Zhao (2000), Shen and Wasserman (2001), and Ghosal, Ghosh and van der Vaart (2000).

Theorem 8 Put independent Normal priors $\theta_i \sim N(0, \tau_i^2)$ where $\tau_i^2 = i^{-2q}$. The Bayes estimator attains the optimal rate only when $q = p + (1/2)$. But then:

$$\pi(\Theta) = 0 \quad \text{and} \quad \pi(\Theta|Y) = 0. \quad (46)$$

7.3 Coverage

Suppose $\pi_n(A) = 1 - \alpha$. Does this imply that

$$\mathbb{P}_{f_0}^n(f_0 \in A) \geq 1 - \alpha? \quad (47)$$

or even

$$\liminf_{n \rightarrow \infty} \inf_{f_0} \mathbb{P}_{f_0}^n(f_0 \in A) \geq 1 - \alpha? \quad (48)$$

Recall what happens for parametric models: if $A = (-\infty, a]$ and

$$\mathbb{P}(\theta \in A | \text{data}) = 1 - \alpha \quad (49)$$

then

$$\mathbb{P}_\theta(\theta \in A) = 1 - \alpha + O\left(\frac{1}{\sqrt{n}}\right) \quad (50)$$

and, moreover, if we use the Jeffreys' prior then

$$\mathbb{P}_\theta(\theta \in A) = 1 - \alpha + O\left(\frac{1}{n}\right). \quad (51)$$

Is the same true for nonparametric models? Unfortunately, no; counterexamples are given by Cox (1993) and Freedman (1999). In their examples, one has:

$$\pi_n(A) = 1 - \alpha \quad (52)$$

but

$$\liminf_{n \rightarrow \infty} \inf_{f_0} \mathbb{P}_{f_0}(f_0 \in A) = 0! \quad (53)$$

8 Bayes Versus Frequentist

People often confuse Bayesian methods and frequentist methods. Bayesian methods are designed for quantifying subjective beliefs. Frequentist methods are designed to create procedures with certain frequency guarantees (consistency, coverage, minimaxity etc). They are two different things. We should use $F(A)$ for frequency and $B(A)$ for belief and then there would be no confusion. Unfortunately, we use the same symbol $P(A)$ for both which causes endless confusion. Let's take a closer look at the differences.

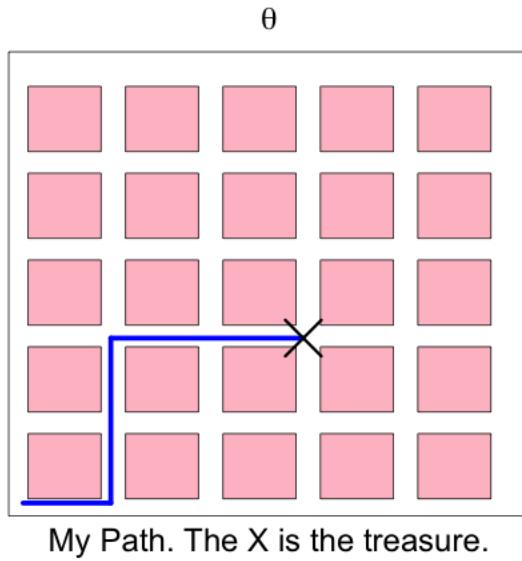
8.1 Adventures in FlatLand: Stone's Paradox

Mervyn Stone is Emeritus Professor at University College London. He is famous for his deep work on Bayesian inference as well as pioneering work on cross-validation, coordinate-free multivariate analysis, as well as many other topics.

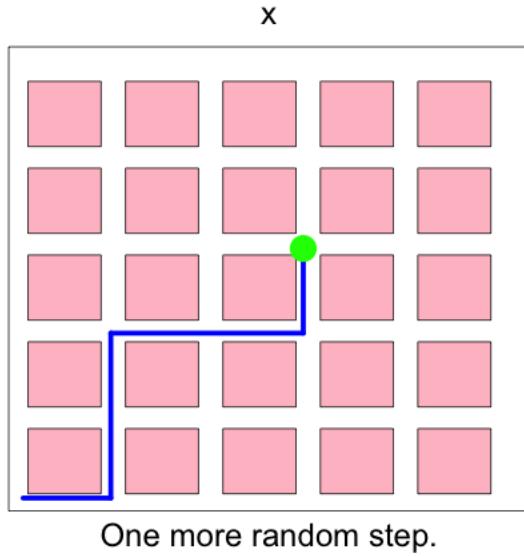
Here we discuss a famous example of his, described in Stone (1970, 1976, 1982). In technical jargon, he shows that “a finitely additive measure on the free group with two generators is nonconglomerable.” In English: even for a simple problem with a discrete parameters space, flat priors can lead to surprises.

Hunting For a Treasure In Flatland. I wonder randomly in a two dimensional grid-world. I drag an elastic string with me. The string is taut: if I back up, the string leaves no slack. I can only move in four directions: North, South, West, East.

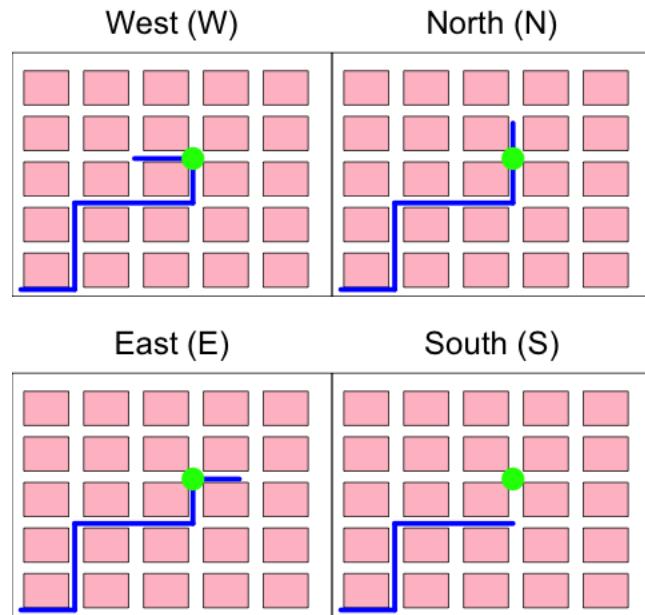
I wander around for a while then I stop and bury a treasure. Call this path θ . Here is an example:



Now I take one more random step. Each direction has equal probability. Call this path x . So it might look like this:



Two people, Bob (a Bayesian) and Carla (a classical statistician) want to find the treasure. There are only four possible paths that could have yielded x , namely:



Let us call these four paths N, S, W, E. The likelihood is the same for each of these. That is, $p(x|\theta) = 1/4$ for $\theta \in \{N, S, W, E\}$. Suppose Bob uses a flat prior. Since the likelihood is

also flat, his posterior is

$$P(\theta = N|x) = P(\theta = S|x) = P(\theta = W|x) = P(\theta = E|x) = \frac{1}{4}.$$

Let B be the three paths that extend x . In this example, $B = \{N, W, E\}$. Then $P(\theta \in B|x) = 3/4$.

Now Carla is very confident and selects a confidence set with only one path, namely, the path that shortens x . In other words, Carla's confidence set is $C = B^c$.

Notice the following strange thing: no matter what θ is, Carla gets the treasure with probability $3/4$ while Bob gets the treasure with probability $1/4$. That is, $P(\theta \in B|x) = 3/4$ but the coverage of B is $1/4$. The coverage of C is $3/4$.

Here is quote from Stone (1976): (except that I changed his B and C to Bob and Carla):

“... it is clear that when Bob and Carla repeatedly engage in this treasure hunt, Bob will find that his posterior probability assignment becomes increasingly discrepant with his proportion of wins and that Carla is, somehow, doing better than [s]he ought. However, there is no message ... that will allow Bob to escape from his Promethean situation; he cannot learn from his experience because each hunt is independent of the other.”

Stone is not criticizing Bayes (as far I can tell). He is just discussing the effect of using a flat prior.

More Trouble For Bob. Let A be the event that the final step reduced the length of the string. Using the posterior above, we see that Bob finds that $P(A|x) = 3/4$ for each x . Since this holds for each x , Bob deduces that $P(A) = 3/4$. On the other hand, Bob notes that $P(A|\theta) = 1/4$ for every θ . Hence, $P(A) = 1/4$. Bob has just proved that $3/4 = 1/4$.

The Source of The Problem. The apparent contradiction stems from the fact that the prior is improper. Technically this is an example of the non-conglomerability of finitely additive measures. For a rigorous explanation of why this happens you should read Stone's papers. Here is an abbreviated explanation, from Kass and Wasserman (1996, Section 4.2.1).

Let π denotes Bob's improper flat prior and let $\pi(\theta|x)$ denote his posterior distribution. Let π_p denote the prior that is uniform on the set of all paths of length p . This is of course a proper prior. For any fixed x , $\pi_p(A|x) \rightarrow 3/4$ as $p \rightarrow \infty$. So Bob can claim that his posterior distribution is a limit of well-defined posterior distributions. However, we need to look at this more closely. Let $m_p(x) = \sum_{\theta} f(x|\theta)\pi_p(\theta)$ be the marginal of x induced by π_p . Let X_p denote all x 's of length p or $p + 1$. When $x \in X_p$, $\pi_p(\theta|x)$ is a poor approximation to $\pi(\theta|x)$ since the former is concentrated on a single point while the latter is concentrated on four points. In fact, the total variation distance between $\pi_p(\theta|x)$ and $\pi(\theta|x)$ is $3/4$ for $x \in X_p$. (Recall that the total variation distance between two probability measures P

and Q is $d(P, Q) = \sup_A |P(A) - Q(A)|$.) Furthermore, X_p is a set with high probability: $m_p(X_p) \rightarrow 2/3$ as $p \rightarrow \infty$.

While $\pi_p(\theta|x)$ converges to $\pi(\theta|x)$ as $p \rightarrow \infty$ for any fixed x , they are not close with high probability. This problem disappears if you use a proper prior. (But that still does not give frequentist coverage.)

The Four Sided Die. Here is another description of the problem. Consider a four sided die whose sides are labeled with the symbols $\{a, b, a^{-1}, b^{-1}\}$. We roll the die several times and we record the label on the lowermost face (there is no uppermost face on a four-sided die). A typical outcome might look like this string of symbols:

$$a \ a \ b \ a^{-1} \ b \ b^{-1} \ b \ a \ a^{-1} \ b$$

Now we apply an annihilation rule. If a and a^{-1} appear next to each other, we eliminate these two symbols. Similarly, if b and b^{-1} appear next to each other, we eliminate those two symbols. So the sequence above gets reduced to:

$$a \ a \ b \ a^{-1} \ b \ b$$

Let us denote the resulting string of symbols, after removing annihilations, by θ . Now we toss the die one more time. We add this last symbol to θ and we apply the annihilation rule once more. This results in a string which we will denote by x .

You get to see x and you want to infer θ .

Having observed x , there are four possible values of θ and each has the same likelihood. For example, suppose $x = (a, a)$. Then θ has to be one of the following:

$$(a), \ (a \ a \ a), \ (a \ a \ b^{-1}), \ (a \ a \ b)$$

The likelihood function is constant over these four values.

Suppose we use a flat prior on θ . Then the posterior is uniform on these four possibilities. Let $C = C(x)$ denote the three values of θ that are longer than x . Then the posterior satisfies

$$P(\theta \in C|x) = 3/4.$$

Thus $C(x)$ is a 75 percent posterior confidence set.

However, the frequentist coverage of $C(x)$ is 1/4. To see this, fix any θ . Now note that $C(x)$ contains θ if and only if θ concatenated with x is smaller than θ . This happens only if the last symbol is annihilated, which occurs with probability 1/4.

Likelihood. Another consequence of Stone's example is that it shows that the Likelihood Principle is bogus. According to the likelihood principle, the observed likelihood function

contains all the useful information in the data. In this example, the likelihood does not distinguish the four possible parameter values. The direction of the string from the current position — which does not affect the likelihood — has lots of information.

Proper Priors. If you want to have some fun, try coming up with proper priors on the set of paths. Then simulate the example, find the posterior and try to find the treasure. If you try this, I'd be interested to hear about the results.

Another question this example raises is: should one use improper priors? Flat priors that do not have a finite sum can be interpreted as finitely additive priors. The father of Bayesian inference, Bruno DeFinetti, was adamant in rejecting the axiom of countable additivity. He thought flat priors like Bob's were fine.

It seems to me that in modern Bayesian inference, there is not universal agreement on whether flat priors are evil or not. But in this example, I think that most statisticians would reject Bob's flat prior-based Bayesian inference.

8.2 The Robins-Ritov Example

This example is due to Robins and Ritov (1997). A simplified version appeared in Wasserman (2004) and Robins and Wasserman (2000). The example is related to ideas from the foundations of survey sampling (Basu 1969, Godambe and Thompson 1976) and also to ancillarity paradoxes (Brown 1990, Foster and George 1996).

Here is (a version of) the example. Consider iid random variables

$$(X_1, Y_1, R_1), \dots, (X_n, Y_n, R_n).$$

The random variables take values as follows:

$$X_i \in [0, 1]^d, \quad Y_i \in \{0, 1\}, \quad R_i \in \{0, 1\}.$$

Think of d as being very, very large. For example, $d = 100,000$ and $n = 1,000$.

The idea is this: we observe X_i . Then we flip a biased coin R_i . If $R_i = 1$ then you get to see Y_i . If $R_i = 0$ then you don't get to see Y_i . The goal is to estimate

$$\psi = P(Y_i = 1).$$

Here are the details. The distribution takes the form

$$p(x, y, r) = p_X(x)p_{Y|X}(y|x)p_{R|X}(r|x).$$

Note that Y and R are independent, given X . For simplicity, we will take $p(x)$ to be uniform on $[0, 1]^d$. Next, let

$$\theta(x) \equiv p_{Y|X}(1|x) = P(Y = 1|X = x)$$

where $\theta(x)$ is a function. That is, $\theta : [0, 1]^d \rightarrow [0, 1]$. Of course,

$$p_{Y|X}(0|x) = P(Y = 0|X = x) = 1 - \theta(x).$$

Similarly, let

$$\pi(x) \equiv p_{R|X}(1|x) = P(R = 1|X = x)$$

where $\pi(x)$ is a function. That is, $\pi : [0, 1]^d \rightarrow [0, 1]$. Of course,

$$p_{R|X}(0|x) = P(R = 0|X = x) = 1 - \pi(x).$$

The function π is **known**. We construct it. Remember that $\pi(x) = P(R = 1|X = x)$ is the probability that we get to observe Y given that $X = x$. Think of Y as something that is expensive to measure. We don't always want to measure it. So we make a random decision about whether to measure it. And we let the probability of measuring Y be a function $\pi(x)$ of x . And we get to construct this function.

Let $\delta > 0$ be a known, small, positive number. We will assume that

$$\pi(x) \geq \delta$$

for all x .

The only thing in the model we don't know is the function $\theta(x)$. Again, we will assume that

$$\delta \leq \theta(x) \leq 1 - \delta.$$

Let Θ denote all measurable functions on $[0, 1]^d$ that satisfy the above conditions. The parameter space is the set of functions Θ .

Let \mathcal{P} be the set of joint distributions of the form

$$p(x) \pi(x)^r (1 - \pi(x))^{1-r} \theta(x)^y (1 - \theta(x))^{1-y}$$

where $p(x) = 1$, and $\pi(\cdot)$ and $\theta(\cdot)$ satisfy the conditions above. So far, we are considering the sub-model \mathcal{P}_π in which π is known.

The parameter of interest is $\psi = P(Y = 1)$. We can write this as

$$\psi = P(Y = 1) = \int_{[0,1]^d} P(Y = 1|X = x)p(x)dx = \int_{[0,1]^d} \theta(x)dx.$$

Hence, ψ is a function of θ . If we know $\theta(\cdot)$ then we can compute ψ .

The usual frequentist estimator is the Horwitz-Thompson estimator

$$\hat{\psi} = \frac{1}{n} \sum_{i=1}^n \frac{Y_i R_i}{\pi(X_i)}.$$

It is easy to verify that $\hat{\psi}$ is unbiased and consistent. Furthermore, $\hat{\psi} - \psi = O_P(n^{-\frac{1}{2}})$. In fact, let us define

$$I_n = [\hat{\psi} - \epsilon_n, \hat{\psi} + \epsilon_n]$$

where

$$\epsilon_n = \sqrt{\frac{1}{2n\delta^2} \log \left(\frac{2}{\alpha} \right)}.$$

It follows from Hoeffding's inequality that

$$\sup_{P \in \mathcal{P}_\pi} P(\psi \in I_n) \geq 1 - \alpha$$

Thus we have a finite sample, $1 - \alpha$ confidence interval with length $O(1/\sqrt{n})$.

Remark: We are mentioning the Horwitz-Thompson estimator because it is simple. In practice, it has three deficiencies:

1. It may exceed 1.
2. It ignores data on the multivariate vector X except for the one dimensional summary $\pi(X)$.
3. It can be very inefficient.

These problems are remedied by using an improved version of the Horwitz-Thompson estimator. One choice is the so-called locally semiparametric efficient regression estimator (Scharfstein et al., 1999):

$$\hat{\psi} = \int \text{expit} \left(\sum_{m=1}^k \hat{\eta}_m \phi_m(x) + \frac{\hat{\omega}}{\pi(x)} \right) dx$$

where $\text{expit}(a) = e^a/(1 + e^a)$, $\phi_m(x)$ are basis functions, and $\hat{\eta}_1, \dots, \hat{\eta}_k, \hat{\omega}$ are the mle's (among subjects with $R_i = 1$) in the model

$$\log \left(\frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} \right) = \sum_{m=1}^k \eta_m \phi_m(x) + \frac{\omega}{\pi(x)}.$$

Here k can increase slowly with n . Recently even more efficient estimators have been derived. Rotnitzky et al (2012) provides a review. In the rest of this post, when we refer to the Horwitz-Thompson estimator, the reader should think "improved Horwitz-Thompson estimator."

To do a Bayesian analysis, we put some prior W on Θ . Next we compute the likelihood function. The likelihood for one observation takes the form $p(x)p(r|x)p(y|x)^r$. The reason

for having r in the exponent is that, if $r = 0$, then y is not observed so the $p(y|x)$ gets left out. The likelihood for n observations is

$$\prod_{i=1}^n p(X_i)p(R_i|X_i)p(Y_i|X_i)^{R_i} = \prod_i \pi(X_i)^{R_i} (1 - \pi(X_i))^{1-R_i} \theta(X_i)^{Y_i R_i} (1 - \theta(X_i))^{(1-Y_i)R_i}.$$

where we used the fact that $p(x) = 1$. But remember, $\pi(x)$ is known. In other words, $\pi(X_i)^{R_i} (1 - \pi(X_i))^{1-R_i}$ is known. So, the likelihood is

$$\mathcal{L}(\theta) \propto \prod_i \theta(X_i)^{Y_i R_i} (1 - \theta(X_i))^{(1-Y_i)R_i}.$$

Combining this likelihood with the prior W creates a posterior distribution on Θ which we will denote by W_n . Since the parameter of interest ψ is a function of θ , the posterior W_n for θ defines a posterior distribution for ψ .

Now comes the interesting part. The likelihood has essentially no information in it.

To see that the likelihood has no information, consider a simpler case where $\theta(x)$ is a function on $[0, 1]$. Now discretize the interval into many small bins. Let B be the number of bins. We can then replace the function θ with a high-dimensional vector $\theta = (\theta_1, \dots, \theta_B)$. With $n < B$, most bins are empty. The data contain no information for most of the θ_j 's. (You might wonder about the effect of putting a smoothness assumption on $\theta(\cdot)$. We'll discuss this in Section 4.)

We should point out that if $\pi(x) = 1/2$ for all x , then Ericson (1969) showed that a certain exchangeable prior gives a posterior that, like the Horwitz-Thompson estimator, converges at rate $O(n^{-1/2})$. However we are interested in the case where $\pi(x)$ is a complex function of x ; then the posterior will fail to concentrate around the true value of ψ . On the other hand, a flexible nonparametric prior will have a posterior essentially equal to the prior and, thus, not concentrate around ψ , whenever the prior W does not depend on the the known function $\pi(\cdot)$. Indeed, we have the following theorem from Robins and Ritov (1997):

Theorem. (Robins and Ritov 1997). Any estimator that is not a function of $\pi(\cdot)$ cannot be uniformly consistent.

This means that, at no finite sample size, will an estimator $\hat{\psi}$ that is not a function of π be close to ψ for all distributions in \mathcal{P} . In fact, the theorem holds for a neighborhood around every pair (π, θ) . Uniformity is important because it links asymptotic behavior to finite sample behavior. But when π is known and is used in the estimator (as in the Horwitz-Thompson estimator and its improved versions) we can have uniform consistency.

Note that a Bayesian will ignore π since the $\pi(X_i)$'s are just constants in the likelihood. There is an exception: the Bayesian can make the posterior be a function of π by choosing a prior W that makes $\theta(\cdot)$ depend on $\pi(\cdot)$. But this seems very forced. Indeed, Robins and

Ritov showed that, under certain conditions, any true subjective Bayesian prior W must be independent of $\pi(\cdot)$. Specifically, they showed that once a subjective Bayesian queries the randomizer (who selected π) about the randomizer's reasoned opinions concerning $\theta(\cdot)$ (but not $\pi(\cdot)$) the Bayesian will have independent priors. We note that a Bayesian can have independent priors even when he believes with probability 1 that $\pi(\cdot)$ and $\theta(\cdot)$ are positively correlated as functions of x i.e. $\int \theta(x) \pi(x) dx > \int \theta(x) dx \int \pi(x) dx$. Having independent priors only means that learning $\pi(\cdot)$ will not change one's beliefs about $\theta(\cdot)$.

9 Freedman's Theorem

Here I will summarize an interesting result by David Freedman (Annals of Mathematical Statistics, Volume 36, Number 2 (1965), 454-456) available at projecteuclid.org.

The result gets very little attention. Most researchers in statistics and machine learning seem to be unaware of the result. The result says that, “almost all” Bayesian posterior distributions are inconsistent, in a sense we’ll make precise below. The math is uncontroversial but, as you might imagine, the interpretation of the result is likely to be controversial.

Let X_1, \dots, X_n be an iid sample from a distribution P on the natural numbers $I = \{1, 2, 3, \dots\}$. Let \mathcal{P} be the set of all such distributions. We endow \mathcal{P} with the weak* topology. Hence, $P_n \rightarrow P$ iff $P_n(i) \rightarrow P(i)$ for all i .

Let μ denote a prior distribution on \mathcal{P} . (More precisely, a prior on an appropriate σ -field.) Let Π be all priors. Again, we endow the set with the weak* topology. Thus $\mu_n \rightarrow \mu$ iff $\int f d\mu_n \rightarrow \int f d\mu$ for all bounded, continuous, real functions f .

Let μ_n be the posterior corresponding to the prior μ after n observations. We will say that the pair (P, μ) is consistent if

$$P^\infty\left(\lim_{n \rightarrow \infty} \mu_n = \delta_P\right) = 1$$

where P^∞ is the product measure corresponding to P and δ_P is a point mass at P .

Now we need to recall some topology. A set is nowhere dense if its closure has an empty interior. A set is meager (or first category) if it is a countable union of nowhere dense sets. Meager sets are small; think of a meager set as the topological version of a null set in measure theory.

Freedman's theorem is: the sets of consistent pairs (P, μ) is meager.

This means that, in a topological sense, consistency is rare for Bayesian procedures. From this result, it can also be shown that most pairs of priors lead to inferences that disagree. (The agreeing pairs are meager.) Or as Freedman says in his paper:

“ ... it is easy to prove that for essentially any pair of Bayesians, each thinks the other is

crazy.”

10 References

- Basu, D. (1969). Role of the Sufficiency and Likelihood Principles in Sample Survey Theory. *Sankya*, 31, 441-454.
- Brown, L.D. (1990). An ancillarity paradox which appears in multiple linear regression. *The Annals of Statistics*, 18, 471-493.
- Ericson, W.A. (1969). Subjective Bayesian models in sampling finite populations. *Journal of the Royal Statistical Society. Series B*, 195-233.
- Foster, D.P. and George, E.I. (1996). A simple ancillarity paradox. *Scandinavian journal of statistics*, 233-242.
- Godambe, V. P., and Thompson, M. E. (1976), Philosophy of Survey-Sampling Practice. In *Foundations of Probability Theory, Statistical Inference and Statistical Theories of Science*, eds. W.L.Harper and A.Hooker, Dordrecht: Reidel.
- Kass, R.E. and Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91, 1343-1370.
- Robins, J.M. and Ritov, Y. (1997). Toward a Curse of Dimensionality Appropriate (CODA) Asymptotic Theory for Semi-parametric Models. *Statistics in Medicine*, 16, 285–319.
- Robins, J. and Wasserman, L. (2000). Conditioning, likelihood, and coherence: a review of some foundational concepts. *Journal of the American Statistical Association*, 95, 1340-1346.
- Rotnitzky, A., Lei, Q., Sued, M. and Robins, J.M. (2012). Improved double-robust estimation in missing data and causal inference models. *Biometrika*, 99, 439-456.
- Scharfstein, D.O., Rotnitzky, A. and Robins, J.M. (1999). Adjusting for nonignorable drop-out using semiparametric nonresponse models. *Journal of the American Statistical Association*, 1096-1120.
- Stone, M. (1970). Necessary and sufficient condition for convergence in probability to invariant posterior distributions. *The Annals of Mathematical Statistics*, 41, 1349-1353,
- Stone, M. (1976). Strong inconsistency from uniform priors. *Journal of the American Statistical Association*, 71, 114-116.
- Stone, M. (1982). Review and analysis of some inconsistencies related to improper priors and finite additivity. *Studies in Logic and the Foundations of Mathematics*, 104, 413-426.

Wasserman, L. (2004). *All of Statistics: a Concise Course in Statistical Inference*. Springer Verlag.

Conformal Prediction

When doing estimation, we usually provide confidence intervals in addition to point estimates. Is there a similar notion for predictions? The answer is yes: we provide *prediction sets* or *set-valued predictions*. Given data $(X_1, Y_1), \dots, (X_n, Y_n)$ we construct a set-valued function C_n , depending on $(X_1, Y_1), \dots, (X_n, Y_n)$ such that

$$P(Y_{n+1} \in C_n(X_{n+1})) \geq 1 - \alpha.$$

The approach we consider in these notes is *conformal prediction*. The idea is due to Vovk, Gammerman and Shafer (2005). The statistical theory for conformal prediction was developed in Lei, Robins and Wasserman (2013), Lei and Wasserman (2014), Lei, G'Sell, Rinaldo, Tibshirani and Wasserman (2017), Sadinle, Lei and Wasserman (2018).

The Unsupervised Case. We begin with the following problem. We observe Y_1, \dots, Y_n and we want to predict Y_{n+1} . The basic algorithm is as follows:

1. Observe Y_1, \dots, Y_n .
2. Define a permutation invariant *residual function* (or *conformity score*) $R_i = \phi(y, \mathcal{A})$ where \mathcal{A} is any dataset of size $n + 1$.
3. For each y :
 - (a) Set $Y_{n+1} = y$ and form the augmented dataset $\mathcal{A} = \{Y_1, \dots, Y_{n+1}\}$.
 - (b) Let $R_i = \phi(Y_i, \mathcal{A})$ for $i = 1, \dots, n + 1$.
 - (c) Test the hypothesis $H_0 : Y_{n+1} = y$ by computing the p-value

$$\pi(y) = \frac{1}{n+1} \sum_{i=1}^{n+1} I(R_i \geq R_{n+1}).$$

- (d) Invert the test: set

$$C_n = \{y : \pi(y) \geq \alpha\}.$$

Note that when H_0 is true, the residuals are exchangeable and the p-value is uniform. Therefore, we have:

Theorem 1 *For every P ,*

$$P(Y_{n+1} \in C_n) \geq 1 - \alpha.$$

If P is absolutely continuous, we also have $P(Y_{n+1} \in C_n) \leq 1 - \alpha + \frac{1}{n+1}$.

Note that this result is distribution-free and holds for all finite samples.

A simple example of a residual function is

$$R_i = \left| Y_i - \frac{Y_1 + \cdots + Y_{n+1}}{n+1} \right|.$$

A more complicated residual is

$$R_i = \frac{1}{\widehat{p}_h(Y_i)}$$

where \widehat{p}_h is a kernel density estimator constructed from the augmented data.

The coverage validity of the prediction set does not depend on the choice of residual. But a poor choice can lead to large prediction sets. A careful choice can lead to minimax optimal sets. For example, suppose that P has a density p . Let t_α be such that $P(Y \in C_*) = 1 - \alpha$ where $C_* = \{y : p(y) \geq t_\alpha\}$. Note that C_* is the smallest set such that $P(Y \in C) = 1 - \alpha$. Suppose that $p \in \text{Holder}(\beta)$ and that there exist c_1, c_2 and γ such that

$$c_1|\epsilon|^\gamma \leq |P(p(Y) \leq t_\alpha + \epsilon) - \epsilon| \leq c_2|\epsilon|^\gamma$$

for all small ϵ . In this case, any prediction set must satisfy $\mu(C_* \Delta C_n) \geq r_n$ with high probability, where μ is Lebesgue, Δ is Lebesgue measure and

$$r_n = \left(\frac{\log n}{n} \right)^{\frac{\beta\gamma}{2\beta+d}}.$$

Theorem 2 *The conformal set C_n based on the kernel density estimator (with appropriate bandwidth) satisfies*

$$P(\mu(C_n \Delta C_\alpha) \geq r_n) \leq \left(\frac{1}{n} \right)^\lambda$$

for any $\lambda > 0$.

For a proof, see Lei, Robins and Wasserman (2013). Thus, in this case, C_n is minimax under the stated conditions. But C_n still has $1 - \alpha$ coverage even if the conditions fail. In fact, C_n has $1 - \alpha$ coverage even if P does not have a density.

The algorithm above requires that we test $H_0 : Y_{n+1} = y$ for every y . In practice, we only consider a grid of values for y . But this can be slow. The *split conformal method* is much faster. The steps are:

1. Split the data into two sets \mathcal{D}_1 and \mathcal{D}_2 .
2. Compute the residuals $R_i = \phi(Y_i, \mathcal{D}_1)$ for $Y_i \in \mathcal{D}_1$.

3. Let q be the $1 - \alpha$ quantile of the residuals.
4. Return $C_n = \{y : \phi(y, \mathcal{D}_1) \leq q\}$.

It is not hard to show that, once again we have

$$P(Y_{n+1} \in C_n) \geq 1 - \alpha$$

for all P . The split conformal method is fast but can result in larger prediction sets. Also, it depends on the particular split of the data. We might consider combining several splits. Suppose that we split the data N times. For each split we construct a prediction set C_j at level $1 - \alpha/N$. It follows that $C = \bigcap_{j=1}^N C_j$. It follows from the union bound that this is valid at level $1 - \alpha$. There are two effects: replacing α with α/N makes each set larger. But taking the intersection makes the set smaller. Unfortunately it can be shown that, under fairly general conditions, that the Lebesgue measure of C is larger than the set constructed with one split, with probability tending to 1. So there seems to be no advantage to suing several splits.

Regression. The extension to regression is straightforward. The data are $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$. We augment the data with a new point (x, y) . Again we define a residual $R_i = \phi((X_i, Y_i), \mathcal{A})$ and we define

$$\pi(x, y) = \frac{1}{n+1} \sum_i I(R_i \geq R_{n+1}).$$

Then we set $C_n(x) = \{y : \pi(x, y) \geq \alpha\}$. We then have

$$P(Y_{n+1} \in C_n(X_{n+1})) \geq 1 - \alpha$$

for every P .

An example of a residual is

$$R_i = |Y_i - \hat{m}(X_i)|$$

where \hat{m} is based on the augmented data. The validity holds even if the model is wrong. Again we can use splitting to speed up the calculations.

Note that the coverage guarantees are marginal. Under regularity conditions it can be shown that we get asymptotic conditional covage, that is,

$$P(Y_{n+1} \in C_n(x) | X_{n+1} = x) \rightarrow 1 - \alpha.$$

It is not possible to get finhite sample, distribution-free conditional coverage as shown on Lei and Wasserman (2014).

We can apply this method to high dimensional and nonparametric regression. The nice thing is that we do not need the model to be correct. To see how well it works, see Figures 1, 2 and 3. (These are from Lei, G'Sell, Rinaldo, Tibshirani and Wasserman 2017.)

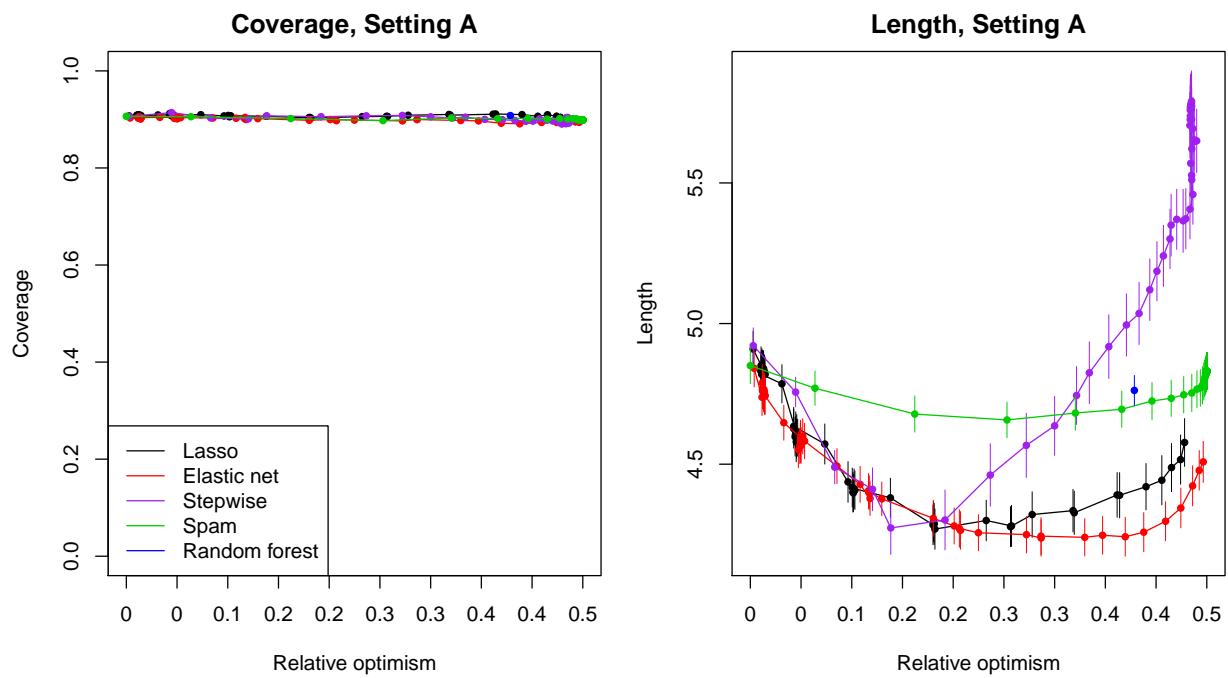


Figure 1: Example: $n = 200, d = 2,000$; linear and Normal

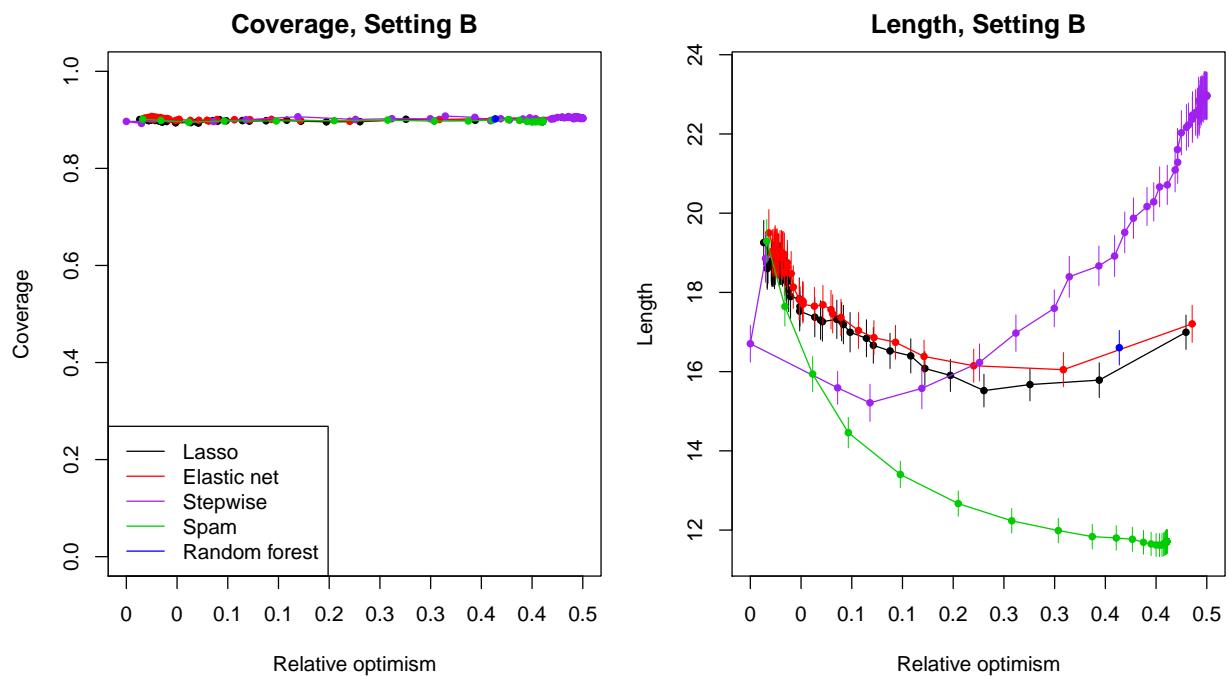


Figure 2: Example: $n = 200, d = 2,000$; nonlinear and heavy-tailed

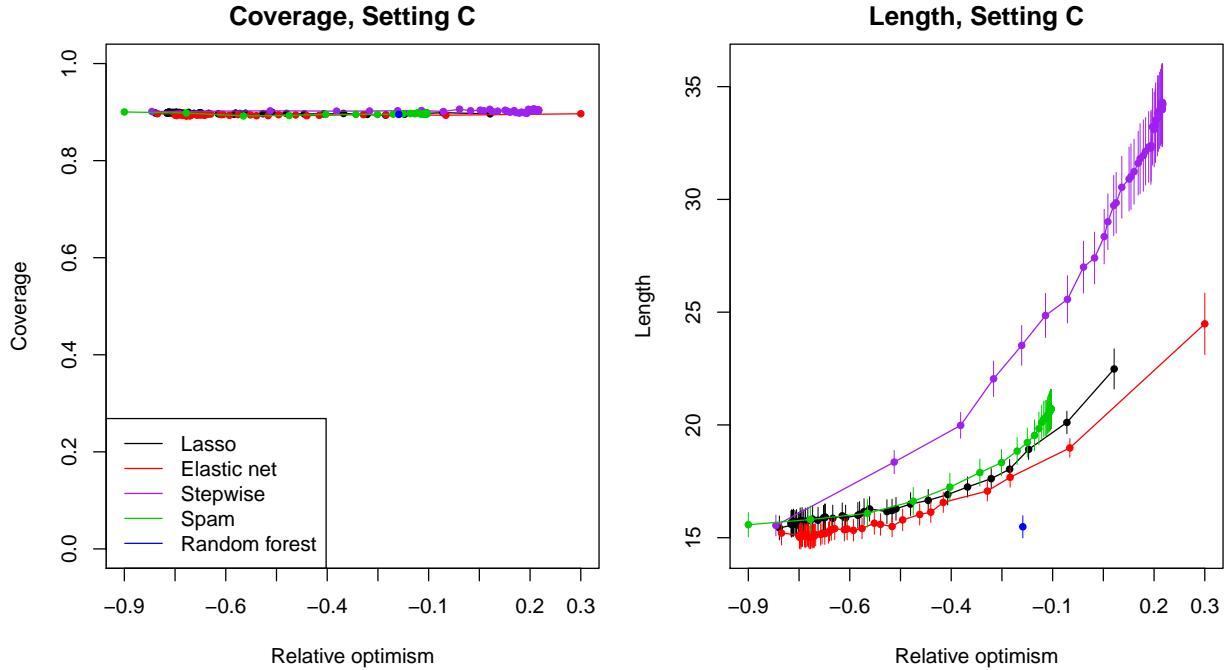


Figure 3: Example: $n = 200, d = 2,000$; linear, correlated, heteroskedastic, heavy-tailed

Classification. The extension to classification is straightforward. The only change is the choice of residual. An example of such a score is $1/\hat{p}(Y_i|X_i)$. Another example is the nearest neighbor score

$$R_i = \frac{\min_{i: Y_i=y} \|x - X_i\|}{\min_{i: Y_i \neq y} \|x - X_i\|}.$$

One complication is that sometimes $C_n(x) = \emptyset$. Some methods for fixing this are discussed in Sadinle, Lei and Wasserman (2018). On the other hand, if one uses the score $1/\hat{p}(X_i|Y_i)$ then $C_n(x) = \emptyset$ when X_i is an outlier i.e. we have not seen a datapoint like X_i before.

Differential Privacy

Protecting privacy while performing statistical analysis is quite challenging. On the one hand, the goal of statistics and machine learning is to be as informative as possible. Protecting privacy is the opposite goal.

How do we formally define privacy? Can we protect privacy and still do an informative analysis? We address these questions in these notes.

1 Introduction

The definition of privacy that has become most common lately is *differential privacy* (Dwork et al 2006).

2 Randomized Response

The predecessor to differential privacy is *randomized response* which is a method used in surveys. It was proposed by Warner in 1965.

I want to know how many students have ever cheated on a test. Suppose that proportion is p . If I ask this question I will not get truthful responses. I tell everyone to flip a coin C with $P(C = 1) = \theta$ and $P(C = 0) = 1 - \theta$. To protect their privacy, I tell them: if the coin is tails answer YES and if the coin is heads answer the question “have you ever cheated?” The observation Y is thus $Y = (1 - C) + CZ$ where $Z = 1$ if they have cheated and $Z = 0$ otherwise. So $\pi \equiv P(Y = 1)$ is $\pi = (1 - \theta) + \theta p$ so that $p = (\pi - 1 + \theta)/\theta$. I can then estimate p by estimating π .

3 Differential Privacy

Suppose we have a dataset X_1, \dots, X_n where $X_i \in \mathcal{X}$. *Knowing the sample space \mathcal{X} explicitly is critical for differential privacy.* The data set $D = \{X_1, \dots, X_n\}$ is in \mathcal{X}^n . Our goal is to report some function $Z = T(D)$ of the data. We will be using some sort of randomization to do this. That is, we will take $Z \sim Q(\cdot | X_1, \dots, X_n)$.

Two datasets D and D' are neighbors if they differ in one random variable. In other words $D = \{X_1, \dots, X_{i-1}, X_i, X_{i+1}, \dots, X_n\}$ and $D' = \{X_1, \dots, X_{i-1}, X'_i, X_{i+1}, \dots, X_n\}$. In this case we write $D \sim D'$.

We say that Q satisfies ϵ -differential privacy if

$$Q(Z \in A|D) \leq e^\epsilon Q(Z \in A|D')$$

for all A and all pairs $D \sim D'$. If Q has density q this means that

$$\sup_z \frac{q(z|D)}{q(z|D')} \leq e^\epsilon.$$

What does the definition mean? It means that whether you are in or not in the database has little affect on the output Z . For example, suppose I think you are person i in the database and I want to guess if your value is $X_i = a$ or $X_i = b$. Before I see any information, suppose my odds are $P(X_i = a)/P(X_i = b)$. After I see Z ,

$$\frac{P(X_i = a|Z)}{P(X_i = b|Z)} = \frac{p(z|X_i = a)}{p(z|X_i = b)} \frac{P(X_i = a)}{P(X_i = b)}$$

and so

$$e^{-\epsilon} \frac{P(X_i = a)}{P(X_i = b)} \leq \frac{P(X_i = a|Z)}{P(X_i = b|Z)} \leq e^\epsilon \frac{P(X_i = a)}{P(X_i = b)}.$$

Since $e^\epsilon \approx 1 + \epsilon$ and ϵ is small, we see that knowing Z does not change my odds much. It is also possible to show that we cannot construct any test with non-trivial power about what your value of X_i is.

So, when differential privacy holds, we cannot learn much about whether a particular person is in the dataset.

4 Queries

The computer science model of privacy is that some curator keeps the data and users send queries about the data. The goal is to release answers that are differentially private.

Let f be a function of the data and define *the sensitivity*

$$\Delta = \sup_{D \sim D'} |f(D) - f(D')|.$$

Suppose we release

$$Z = f(D) + W$$

where $f(w) \propto e^{-w/\lambda}$. Note that W has a Laplace distribution with standard deviation $\sqrt{2}\lambda$. If we set $\lambda = \Delta/\epsilon$ then

$$\frac{p(z|D)}{p(z|D')} \leq e^{|f(D) - f(D')|/\lambda} \leq e^\epsilon$$

so that differential privacy holds.

For example, suppose that $X_1, \dots, X_n \in [-B, B]$ and that $f(D) = \bar{X}$. Then $\Delta = 2B/n$ so we need to add noise with standard deviation $O(B/(n\epsilon))$. As a function of n this is good. As a function of B it is bad.

5 How Informative is Z ?

Obviously we lose information when we use differential privacy.

As an extreme example, suppose the data are on $[0, 1]$ and suppose the true distribution F is a point mass at $x \in [0, 1]$. So the dataset is $X = (X_1, \dots, X_n) = (x, x, \dots, x)$. Suppose we output Z_1, \dots, Z_k from a differentially private $Q(Z|X)$. Let \hat{F} be the empirical distribution of Z . Then it can be shown that \hat{F} must be inconsistent, that is, there exists $\delta > 0$ such that,

$$\liminf_{n \rightarrow \infty} P\left(\sup_s |F(s) - \hat{F}(s)| > \delta\right) > 0.$$

See Blum, Ligett and Roth (2008) and Wasserman and Zhou (2010).

Barber and Duchi (2014) studied differential privacy from the minimax point of view. Suppose we observe $X_1, \dots, X_n \sim P$ where $X_i \in [0, 1]^d$. Consider the simple task of estimating the mean μ . If we ignore privacy, then we can use \bar{X} which is risk $\mathbb{E}[||\bar{X} - \mu||^2] \leq d/n$. They showed that any differentially private estimator $\tilde{\mu}$ satisfies the lower bound

$$\mathbb{E}[||\tilde{\mu} - \mu||^2] \geq \frac{d}{n} + \frac{d^3}{n^2\epsilon^2} = \frac{d}{n} \left[1 + \frac{d^2}{n\epsilon^2} \right].$$

So the price we pay for privacy is $\frac{d^3}{n^2\epsilon^2}$ which is quite steep.

6 Releasing a Whole Dataset

Statisticians have been less enthusiastic about differential privacy than computer scientists. One of the reasons for this is the heavy dependence on the notion of using privatized queries. The idea that we would analyze data by sending queries to a curator is unrealistic. Real data analysis involves: looking at the data, fitting models, testing fit, making predictions, constructing confidence sets etc. This requires access to the whole data set. This leads to the following questions. Can we release a privatized version of the whole dataset? In fact, there are several ways to do this.

6.1 Exponential Mechanism

The exponential mechanism, due to McSherry and Talwar (2007), is a general method for preserving differential privacy. Here, I'll discuss the special case where we want to release a private data set $Z = (Z_1, \dots, Z_k)$. Let $\xi(x, y)$ be some function that measures the distance between two data sets $x = (x_1, \dots, x_n)$ and $z = (z_1, \dots, z_k)$. Define the sensitivity

$$\Delta = \sup_{x \sim y} \sup_z |\xi(x, z) - \xi(y, z)|.$$

Now draw $Z = (Z_1, \dots, Z_k)$ from the density

$$q(z|x) \propto \exp\left(-\frac{\epsilon \xi(x, z)}{2\Delta}\right).$$

It is easy to check that this satisfies ϵ -differential privacy.

As an example, suppose that \mathcal{X} is compact and define $\xi(x, z) = \sup_t |F_x(t) - F_z(t)| = \|F_x - F_z\|_\infty$ where F_x is the empirical cdf of $x = (x_1, \dots, x_n)$ and F_z is the empirical cdf of $z = (z_1, \dots, z_k)$. So ξ is the Kolmogorov-Smirnov distance. In this case, $\Delta = 1/n$ and so we draw z from the density

$$q(z|x) \propto \exp\left(-\frac{n\epsilon \|F_x - F_z\|_\infty}{2}\right).$$

Wasserman and Zhou (2010) showed that, for this scheme, the optimal choice of k is $k \asymp n^{2/3}$ and that $\|F - F_z\|_\infty = O_P(n^{-1/3})$. Without privacy we have $\|F - F_x\|_\infty = O_P(n^{-1/2})$. So we see that we have lost accuracy.

More generally, they showed that

$$P(\|F - F_z\|_\infty > \delta) \leq \frac{(\sup_x p(x))^k e^{-3\epsilon\delta n/16}}{S(k, \delta/2)}$$

where $S(k, \delta/2)$ is the *small ball probability*, that is $P(\|F - F_z\| \leq \delta/2)$. However, it is not known if these bound are tight.

6.2 Density Estimation I

Another way to release a privatized dataset is to compute a privatized density estimate \hat{p} . Then we can draw a sample $Z_1, \dots, Z_N \sim \hat{p}$. It is easy to show that if \hat{p} is differentially private then so is $Z = (Z_1, \dots, Z_N)$.

Dwork et al (2006) suggested using a privatized histogram which was analyzed in Wasserman and Zhou (2010). Suppose that the data are on $[0, 1]^d$. Divide the space into $m = 1/h^d$ bins

B_1, \dots, B_m and form the usual histogram estimator

$$\hat{p}(x) = \sum_j \frac{\hat{p}_j}{h^d} I(x \in B_j)$$

where $\hat{p}_j = C_j/n$ and C_j is the number of observations in bin B_j . To privatize \hat{p} , define

$$\hat{q}(x) = \sum_j \frac{\hat{q}_j}{h^d} I(x \in B_j)$$

where $\hat{q}_j = \tilde{D}_j / \sum_s \tilde{D}_s$, $\tilde{D}_j = \max\{D_j, 0\}$ and $D_j = C_j + \nu_j$ where ν_j is drawn from a Laplace density with mean 0 and variance $8/\epsilon^2$. Wasserman and Zhou (2010) showed that, if X has a Lipschitz density, and if we choose $m \asymp n^{d/(2+d)}$ the histogram of the privatized density achieves the minimax rate $n^{-2/(2+d)}$. So in this case, there is no loss in rate by releasing the privatized data or the privatized histogram.

However, the minimax rate is not the whole story. Suppose that the original histogram \hat{p} is sparse i.e. has many empty cells. The privatized histogram \hat{q} is forced to “fill in” these empty cells. So in these cases, \hat{q} will look very different from \hat{p} . In particular, much of the clustering structure will be lost. And if there is any lower dimensional structure in the data, this will be destroyed.

6.3 Density Estimation II

A second approach is based on orthogonal series. For simplicity assume that $\mathcal{X} = [0, 1]$. Write

$$p(x) = 1 + \sum_{j=1}^{\infty} \beta_j \psi_j(x)$$

where $\{1, \psi_1, \psi_2, \dots\}$ is an orthonormal basis. Suppose that $\sum_j \beta_j^2 j^{2\gamma} \leq C^2$. This is a Sobolev ellipsoid. The minimax rate is $n^{-2\gamma/(2\gamma+1)}$.

The usual density estimator in this framework is

$$\hat{p}(x) = 1 + \sum_{j=1}^m \hat{\beta}_j \psi_j(x)$$

where $m = n^{1/(2\gamma+1)}$ and $\hat{\beta}_j = n^{-1} \sum_i \psi_j(X_i)$ which achieves the minimax rate. A privatized estimator is

$$\hat{q}(x) = 1 + \sum_{j=1}^m (\hat{\beta}_j + \nu_j) \psi_j(x)$$

where ν_j is Laplace with mean 0 and standard deviation $mc_0/(n\epsilon)$ where $c_0 = \sup_j \sup_x |\psi_j(x)|$. It turns out that \hat{q} also achieves the minimax rate.

6.4 Density Estimation III

The most commonly used density estimator is the kernel density estimator

$$\widehat{p}(x) = \frac{1}{n} \sum_i \frac{1}{h^d} K\left(\frac{x - X_i}{h}\right).$$

Is there a way to privatize \widehat{p} ?

This is trickier than histograms and orthogonal series estimators since \widehat{p} is not easily described by a finite set of parameters. In principle, we want to draw a random function g such that $P(g \in A|D) \leq e^\epsilon P(g \in A|D')$. But the sets A are now subsets of some function space and it is not immediately clear how to do this.

So far, I know of only two ways to do this. Hall, Rinaldo and Wasserman (2013) suggested using

$$g = \widehat{p} + \frac{C}{n\epsilon h^{d/2}} G$$

where C is an appropriate constant and G is a mean 0 Gaussian process with a certain covariance structure. The resulting density estimator g is very wiggly but it does satisfy differential privacy. Moreover, it has the same rate of convergence as the original density estimator.

A second approach was recently presented by Alda and Rubinstein (2017). The first create a grid on the sample space. Next, the approximate \widehat{p} using Bernstein polynomials. Then the add Laplace noise to the coefficients of the polynomials.

I should add that the methods in Hall, Rinaldo and Wasserman (2013) and Alda and Rubinstein (2017) are quite general and can be used for the private release of fairly general functions. In fact, Alda and Rubinstein (2017) also apply their approach to classification, logistic regression and empirical risk minimization. They also provide a lower bound which shows that we must, in general, introduce an error of size Δ/ϵ when privately releasing a function, where Δ is the sensitivity defined earlier.

7 Conclusion

Differential privacy (DP) is a very active area of research. Here is a summary of the strengths and weaknesses of this approach:

Strengths:

1. DP gives a very rigorous, precise notion of privacy.

2. Many methods in machine learning and statistics can be made differentially private.
3. DP can be used for other purposes. For example, Dwork et al (2015) created a method called *reusable holdout* that allows an interactive approach to data analysis while making repeated looks at the data without introducing too much bias. The goal of the method is to impose a sort of differential privacy on each step of the analysis.

Weaknesses:

1. DP has dominated the research in privacy. It seems that there is not much research in other approaches.
2. DP is very strong. You need to add a lot of noise to the data.
3. When there is a structure in the data, such as voids, manifolds etc, it is destroyed by DP.
4. I have not seen it really used in much practical data analysis.

Optimal Transport and Wasserstein Distance

The Wasserstein distance — which arises from the idea of *optimal transport* — is being used more and more in Statistics and Machine Learning. In these notes we review some of the basics about this topic. Two good references for this topic are:

Kolouri, Soheil, et al. Optimal Mass Transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine* 34.4 (2017): 43-59.

Villani, Cedric. *Topics in optimal transportation*. No. 58. American Mathematical Soc., 2003.

As usual, you can find a wealth of information on the web.

1 Introduction

Let $X \sim P$ and $Y \sim Q$ and let the densities be p and q . We assume that $X, Y \in \mathbb{R}^d$. We have already seen that there are many ways to define a distance between P and Q such as:

$$\begin{aligned}\text{Total Variation : } & \sup_A |P(A) - Q(A)| = \frac{1}{2} \int |p - q| \\ \text{Hellinger : } & \sqrt{\int (\sqrt{p} - \sqrt{q})^2} \\ L_2 : & \int (p - q)^2 \\ \chi^2 : & \int \frac{(p - q)^2}{q}.\end{aligned}$$

These distances are all useful, but they have some drawbacks:

1. We cannot use them to compare P and Q when one is discrete and the other is continuous. For example, suppose that P is uniform on $[0, 1]$ and that Q is uniform on the finite set $\{0, 1/N, 2/N, \dots, 1\}$. Practically speaking, there is little difference between these distributions. But the total variation distance is 1 (which is the largest the distance can be). The Wasserstein distance is $1/N$ which seems quite reasonable.
2. These distances ignore the underlying geometry of the space. To see this consider Figure 1. In this figure we see three densities p_1, p_2, p_3 . It is easy to see that $\int |p_1 - p_2| = \int |p_1 - p_3| = \int |p_2 - p_3|$ and similarly for the other distances. But our intuition tells us that p_1 and p_2 are close together. We shall see that this is captured by Wasserstein distance.

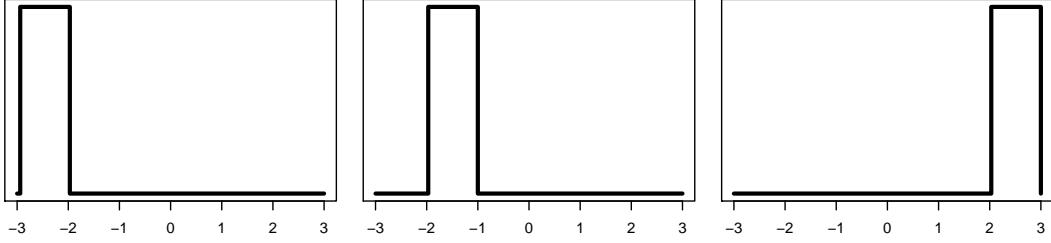


Figure 1: *Three densities p_1, p_2, p_3 . Each pair has the same distance in L_1 , L_2 , Hellinger etc. But in Wasserstein distance, p_1 and p_2 are close.*

3. When we average different objects — such as distributions or images — we would like to make sure that we get back a similar object. The top plot in Figure 2 shows some distributions, each of which is uniform on a circle. The bottom left plot shows the Euclidean average of the distributions which is just a gray mess. The bottom right shows the Wasserstein barycenter (which we will define later) which is a much better summary of the set of images.
4. When we compute the usual distance between two distributions, we get a number but we don't get any qualitative information about why the distributions differ. But with the Wasserstein distance we also get a map that shows us how we have to move the mass of P to morph it into Q .
5. Suppose we want to create a path of distributions (a geodesic) P_t that interpolates between two distributions P_0 and P_1 . We would like the distributions P_t to preserve the basic structure of the distributions. Figure 5 shows an example. The top row shows the path between P_0 and P_1 using Wasserstein distance. The bottom row shows the path using L_2 distance. We see that the Wasserstein path does a better job of preserving the structure.
6. Some of these distances are sensitive to small wiggles in the distribution. But we shall see that the Wasserstein distance is insensitive to small wiggles. For example if P is uniform on $[0, 1]$ and Q has density $1 + \sin(2\pi kx)$ on $[0, 1]$ then the Wasserstein distance is $O(1/k)$.

2 Optimal Transport

If $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ then the the distribution of $T(X)$ is called the push-forward of P , denoted by $T_\#P$. In other words,

$$T_\#P(A) = P(\{x : T(x) \in A\}) = P(T^{-1}(A)).$$

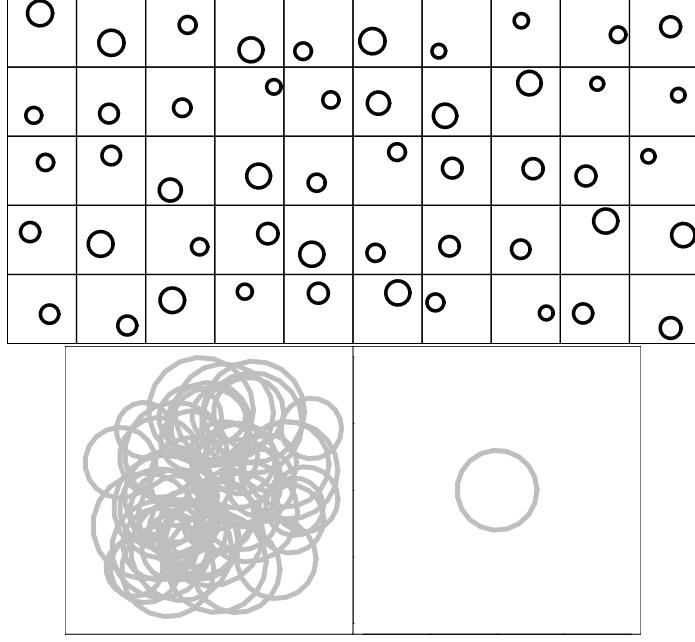


Figure 2: *Top:* Some random circles. *Bottom left:* Euclidean average of the circles. *Bottom right:* Wasserstein barycenter.

The *Monge* version of the optimal transport distance is

$$\inf_T \int ||x - T(x)||^p dP(x)$$

where the infimum is over all T such that $T_{\#}P = Q$. Intuitively, this measures how far you have to move the mass of P to turn it into Q . A minimizer T^* , if one exists, is called the *optimal transport map*.

If P and Q both have densities than T^* exists. The map $T_t(x) = (1-t)x + tT^*(x)$ gives the path of a particle of mass at x . Also, $P_t = T_{t\#}P$ is the geodesic connecting P to Q .

But, the minimizer might not exist. Consider $P = \delta_0$ and $Q = (1/2)\delta_{-1} + (1/2)\delta_1$ where δ_a . In this case, there is no map T such that $T_{\#}P = Q$. This leads us to the Kantorovich formulation where we allow the mass at x to be split and move to more than one location.

Let $\mathcal{J}(P, Q)$ denote all joint distributions J for (X, Y) that have marginals P and Q . In other words, $T_X J = P$ and $T_Y J = Q$ where $T_X(x, y) = x$ and $T_Y(x, y) = y$. Figure 4 shows an example of a joint distribution with two given marginal distributions. Then the Kantorovich, or Wasserstein, distance is

$$W_p(P, Q) = \left(\inf_{J \in \mathcal{J}(P, Q)} \int ||x - y||^p dJ(x, y) \right)^{1/p}$$

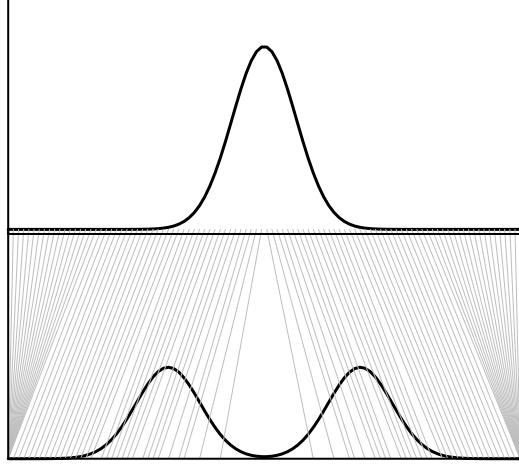


Figure 3: Two densities p and q and the optimal transport map to that morphs p into q .

where $p \geq 1$. When $p = 1$ this is also called the *Earth Mover distance*. The minimizer J^* (which does exist) is called the *optimal transport plan* or the *optimal coupling*. In case there is an optimal transport map T then J is a singular measure with all its mass on the set $\{(x, T(x))\}$.

It can be shown that

$$W_p^p(P, Q) = \sup_{\psi, \phi} \int \psi(y) dQ(y) - \int \phi(x) dP(x)$$

where $\psi(y) - \phi(x) \leq \|x - y\|^p$. This is called the dual formulation. In special case where $p = 1$ we have the very simple representation

$$W_1(P, Q) = \sup \left\{ \int f(x) dP(x) - \int f(x) dQ(x) : f \in \mathcal{F} \right\}$$

where \mathcal{F} denotes all maps from \mathbb{R}^d to \mathbb{R} such that $|f(y) - f(x)| \leq \|x - y\|$ for all x, y .

When $d = 1$, the distance has a closed form:

$$W_p(P, Q) = \left(\int_0^1 |F^{-1}(z) - G^{-1}(z)|^p dz \right)^{1/p}$$

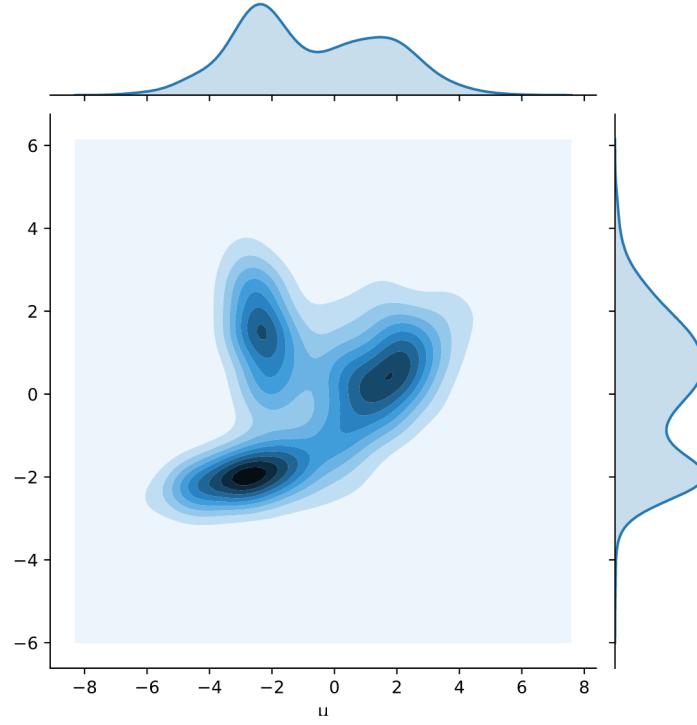


Figure 4: This plot shows one joint distribution J with a given X marginal and a given Y marginal. Generally, there are many such joint distributions. Image credit: Wikipedia.

and F and G are the cdf's of P and Q . If P is the empirical distribution of a dataset X_1, \dots, X_n and Q is the empirical distribution of another dataset Y_1, \dots, Y_n of the same size, then the distance takes a very simple function of the order statistics:

$$W_p(P, Q) = \left(\sum_{i=1}^n \|X_{(i)} - Y_{(i)}\|^p \right)^{1/p}.$$

An interesting special case occurs for Normal distributions. If $P = N(\mu_1, \Sigma_1)$ and $Q = N(\mu_2, \Sigma_2)$ then

$$W^2(P, Q) = \|\mu_1 - \mu_2\|^2 + B^2(\Sigma_1, \Sigma_2)$$

where

$$B^2(\Sigma_1, \Sigma_2) = \text{tr}(\Sigma_1) + \text{tr}(\Sigma_2) - 2\text{tr}\left[(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\right].$$

There is a connection between Wasserstein distance and L_1 distance (Indyk and Thaper 2003). Suppose that P and Q are supported on $[0, 1]^d$. Let G_1, G_2, \dots be a dyadic sequence of cubic partitions where each cube in G_i has side length $1/2^i$. Let $p^{(i)}$ and $q^{(i)}$ be the multinomials from P and Q one grid G_i . Fix $\epsilon > 0$ and let $k = \log(2d/\epsilon)$. Then

$$W_1(P, Q) \leq 2d \sum_{i=1}^m \frac{1}{2^i} \|p^{(i)} - q^{(i)}\|_1 + \frac{\epsilon}{2}. \quad (1)$$

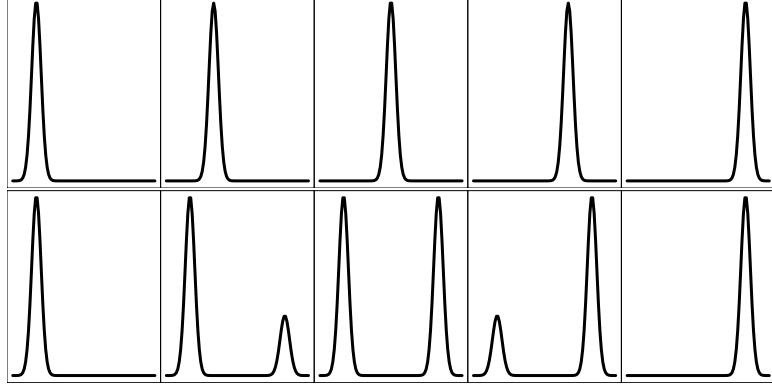


Figure 5: Top row: Geodesic path from P_0 to P_1 . Bottom row: Euclidean path from P_0 to P_1 .

There is an almost matching lower bound (but it actually requires using a random grid).

More generally, as discussed in Weed and Bach (2017), for any sequence of dyadic partitions $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m$ we have

$$W_p^p(P, Q) \leq \delta^{mp} + \sum_{j=1}^m \delta^{(j-1)p} \sum_{A \in \mathcal{A}_j} |P(A) - Q(A)|$$

where $\text{diam}(A) \leq \delta^j$ for every $A \in \mathcal{A}_j$.

These results show that, in some sense, Wasserstein distance is like a multiresolution L_1 distance.

3 Geodesics

Let P_0 and P_1 be two distributions. Consider a map c taking $[0, 1]$ to the set of distributions, such that $c(0) = P_0$ and $c(1) = P_1$. Thus $(P_t : 0 \leq t \leq 1)$ is a path connecting P_0 and P_1 , where $P_t = c(t)$. The length of c — denoted by $L(c)$ — is the supremum of $\sum_{i=1}^m W_p(c(t_{i-1}), c(t_i))$ over all m and all $0 = t_1 < \dots < t_m = 1$. There exists such a path c such that $L(c) = W(P_0, P_1)$. In other words, $(P_t : 0 \leq t \leq 1)$ is the geodesic connecting P_0 and P_1 . It can be shown that

$$P_t = F_{t\#} J$$

where J is the optimal coupling and $F_t(x, y) = (1-t)x + ty$. Examples are shown in Figures 5 and 6.

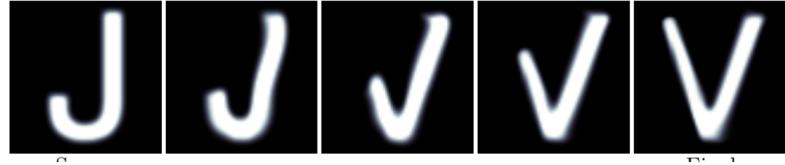


Figure 6: *Morphing one image into another using the Wasserstein geodesic.* Image credit: Bauer, Joshi and Modin 2015.

4 Barycenters and PCA

Suppose we have a set of distributions P_1, \dots, P_N . How do we summarize these distributions with one “typical” distribution? We could take the average $\frac{1}{N} \sum_{j=1}^n P_j$. But the resulting average won’t look like any of the P_j ’s. See Figure 7.

Instead we can use the Wasserstein barycenter which is the distribution P that minimizes

$$\sum_{j=1}^N W(P, P_j).$$

The bottom right plot of Figure 7 shows an example. You can see that this does a much better job.

We can do the same thing for data sets. See Figure 8. Here we simple regard a dataset as an empirical distribution. The average (red dots) $N^{-1} \sum_j \hat{P}_j$ of these empirical distributions \hat{P}_j is useless. But the Wasserstein barycenter (blue dots) gives us a better sense of what a typical dataset looks like.

Let’s pursue this last example a bit more since it will give us some intuition. Suppose we have N datasets $\mathcal{X}_1, \dots, \mathcal{X}_N$ where $\mathcal{X}_j = \{X_{j1}, \dots, X_{jn}\}$. For simplicity, suppose that each is of the same size n . In this case, we can describe the Wasserstein barycenter in a simple way. First we find the order statistics for each data set:

$$X_{(j1)} \leq X_{(j2)} \leq \dots \leq X_{(jn)}.$$

Now for each $1 \leq r \leq n$, we find the average r^{th} average order statistic:

$$Y_{(r)} = \frac{1}{N} \sum_{j=1}^N X_{(jr)}.$$

Then $\mathcal{Y} = \{Y_{(1)}, \dots, Y_{(n)}\}$ is the Wasserstein barycenter. In a sense, all we are really doing is converting to quantiles and averaging.

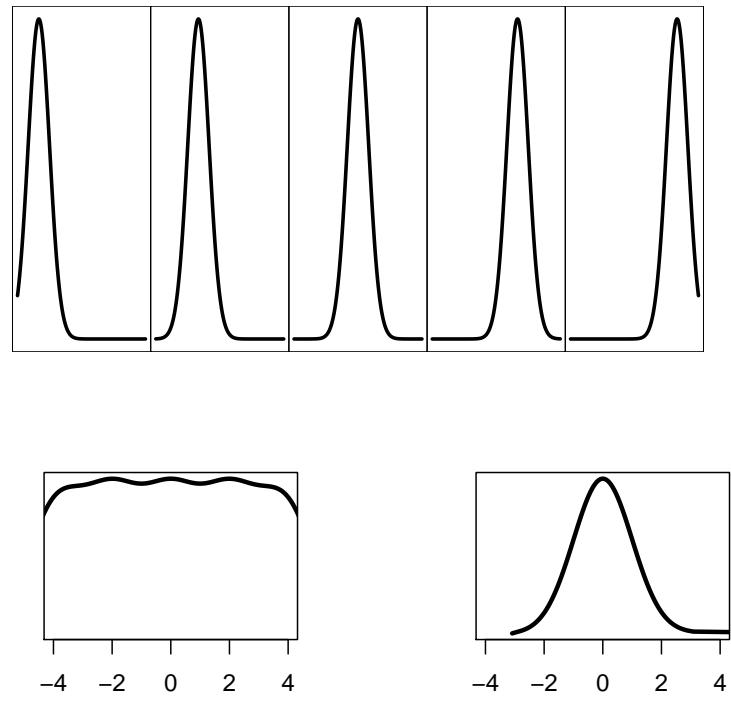


Figure 7: *Top: Five distributions. Bottom left: Euclidean average of the distributions. Bottom right: Wasserstein barycenter.*

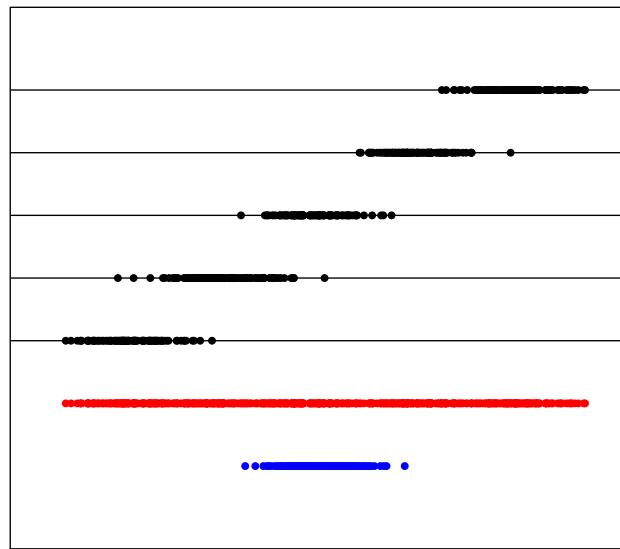


Figure 8: *The top five lines show five, one-dimensional datasets. The red points the what happens if we simple average the give empirical distributions. The blue dots show the Wasserstein barycenter which, in this case, can be obtained simply by averaging the order statistics.*

If $P_j = N(\mu_j, \Sigma_j)$ for $j = 1, \dots, N$ then the Barycenter is $N(\mu, \Sigma)$ where $\mu = N^{-1} \sum_j \mu_j$ and Σ satisfies

$$\frac{1}{N} \sum_j (\Sigma^{1/2} \Sigma_j \Sigma^{1/2})^{1/2}.$$

Now that we have a notion of average, it is possible to define a Wasserstein version of PCA. There are several approaches; see, for example Seguy and Cuturi (2015), Boissard et al (2013), Bigot (2014), Wang, Wei and Slepcev (2013). The idea, as with the barycenters, is to find orthogonal directions of variation in the space of measures (or images). Here I'll briefly describe the method from Wang, Wei and Slepcev (2013).

Let P_1, \dots, P_N be distributions with densities. Let R be a reference distribution with density r . Define $\psi_j(x) = (T_j(x) - x)\sqrt{r(x)}$. The set of distributions endowed with the W^2 distance is a manifold and $\int (\psi_j(x) - \psi_k(x))^2 dx$ is the distance between the projections onto the tangent space at R . In other words, ψ_j defines an approximate embedding of the set of distributions and L^2 . We can now perform PCA on the functions ψ_1, \dots, ψ_N .

5 Minimax Rates

Equation (1) can be used to compute rates of convergence. Suppose that the sample space is $[0, 1]^d$. The minimax rate is (ignoring log factors)

$$\epsilon_n \asymp \begin{cases} n^{-1/(2p)} & p \geq d/2 \\ n^{-1/d} & p < d/2. \end{cases}$$

The optimal estimator is the empirical distribution. This is a nice property about Wasserstein: there is no need to smooth.

Now suppose we observe $X_1, \dots, X_n \sim P$ supported on $[0, \Delta]^d$. We want to test $H_0 : P = P_0$ versus $H_1 : W_1(P, P_0) > \epsilon$. Ba et al (2013) and Deng, Li and Wu (2017) showed that the minimax separation rate is (ignoring some log terms)

$$\epsilon_n \asymp \frac{2\Delta d}{n^{\frac{3}{2d}}}.$$

In the special case were P and P_0 are concentrated in k small clusters, the rate becomes

$$\epsilon_n \asymp d\Delta \left(\frac{k}{n} \right)^{1/4}.$$

6 Confidence Intervals

How do we get hypothesis tests and confidence intervals for the Wasserstein distance? Usually, we would use some sort of central limit theorem. Such results are available when $d = 1$ but are elusive in general.

del Barrio and Loubes (2017) show that

$$\sqrt{n}(W_2^2(P, P_n) - \mathbb{E}[W_2^2(P, P_n)]) \rightsquigarrow N(0, \sigma^2(P))$$

for some $\sigma^2(P)$. And, in the two sample case

$$\sqrt{\frac{nm}{n+m}} \left(W_2^2(P_n, Q_m) - \mathbb{E}[W_2^2(P_n, Q_m)] \right) \rightsquigarrow N(0, \sigma^2(P, Q))$$

for some $\sigma^2(P, Q)$. Unfortunately, these results do not give a confidence interval for $W(P, Q)$ since the limit is centered around $\mathbb{E}[W_2^2(P_n, Q_m)]$ instead of $W_2^2(P, Q)$. However, del Barrio, Gordaliza and Loubes (2018) show that if some smoothness assumptions holds, then the distribution centers around $W_2^2(P, Q)$. More generally, Tudor, Siva and Larry have a finite sample confidence interval for $W(P, Q)$ without any conditions.

All this is for $d = 1$. The case $d > 1$ seems to be unsolved.

Another interesting case is when the support $\mathcal{X} = \{x_1, \dots, x_k\}$ is a finite metric space. In this case, Sommerfeld and Munk (2017) obtained some precise results. First, they showed that

$$\left(\frac{nm}{n+m} \right)^{\frac{1}{2p}} W_p(P_n, Q_m) \rightsquigarrow \left(\max_u \langle G, u \rangle \right)^{1/p}$$

G is a mean 0 Gaussian random vector and u varies over a convex set. By itself, this does not yield a confidence set. But they showed that the distribution can be approximated by subsampling, where the subsamples of size m with $m \rightarrow \infty$ and $m = o(n)$.

You might wonder why the usual bootstrap does not work. The reason is that the map $(P, Q) \mapsto W_p^p(P, Q)$ is not Hadamard differentiable. This means that the map does not have smooth derivatives. In general, the problem of constructing confidence intervals for Wasserstein distance is unsolved.

7 Robustness

One problem with the Wasserstein distance is that it is not robust. To see this, note that $W(P, (1 - \epsilon)P + \epsilon\delta_x) \rightarrow \infty$ as $x \rightarrow \infty$.

However, a partial solution to the robustness problem is available due to Alvarez-Esteban, del Barrio, Cuesta Albertos and Matran (2008). They define the α -trimmed Wasserstein distance

$$\tau(P, Q) = \inf_A W_2(P_A, Q_A)$$

where $P_A(\cdot) = P(A \cap \cdot)/P(A)$, $Q_A(\cdot) = Q(A \cap \cdot)/Q(A)$ and A varies over all sets such that $P(A) \geq 1 - \alpha$ and $Q(A) \geq 1 - \alpha$. When $d = 1$, they show that

$$\tau(P, Q) = \inf_A \left(\frac{1}{1 - \alpha} \int_A (F^{-1}(t) - G^{-1}(t))^2 dt \right)^{1/2}$$

where A varies over all sets with Lebesgue measure $1 - \alpha$.

8 Inference From Simulations

Suppose we have a parametric model $(P_\theta : \theta \in \Theta)$. We can estimate θ using the likelihood function $\prod_i p_\theta(X_i)$. But in some cases we cannot actually evaluate p_θ . Instead, we can simulate from P_θ . This happens quite often, for example, in astronomy and climate science. Berntom et al (2017) suggest replacing maximum likelihood with minimum Wasserstein distance. That is, given data X_1, \dots, X_n we use

$$\hat{\theta} = \operatorname{argmin}_\theta W(P_\theta, P_n)$$

where P_n is the empirical measure. We estimate $W(P_\theta, P_n)$ by $W(Q_N, P_n)$ where Q_N is the empirical measure based on a sample $Z_1, \dots, Z_N \sim P_\theta$.

9 Computing the Distance

We saw that, when $d = 1$,

$$W_p(P, Q) = \left(\int_0^1 |F^{-1}(z) - G^{-1}(z)|^p dz \right)^{1/p}$$

and F and G are the cdf's of P and Q . If P is the empirical distribution of a dataset X_1, \dots, X_n and Q is the empirical distribution of another dataset Y_1, \dots, Y_n of the same size, then the distance takes a very simple function of the order statistics:

$$W_p(P, Q) = \left(\sum_{i=1}^n ||X_{(i)} - Y_{(i)}||^p \right)^{1/p}.$$

The one dimensional case is, perhaps, the only case where computing W is easy.

For any d , if P and Q are empirical distributions — each based on n observations — then

$$W_p(P, Q) = \inf_{\pi} \left(\sum_i \|X_i - Y_{\pi(i)}\|^p \right)^{1/p}$$

where the infimum is over all permutations π . This may be solved in $O(n^3)$ time using the Hungarian algorithm.

Suppose that P has density p and that $Q = \sum_{j=1}^m q_j \delta_{y_j}$ is discrete. Given weights $w = (w_1, \dots, w_m)$ define the power diagram V_1, \dots, V_m where $y \in V_j$ if y is closer to the ball $B(y_j, w_j)$ than any other ball $B(y_s, w_s)$. Define the map $T(x) = y_j$ when $x \in V_j$. According to a result known as Bernier's theorem, if have that $P(V_j) = q_j$ then

$$W_2(P, Q) = \left(\sum_j \int_{V_j} \|x - y_j\|^2 dP(x) \right)^{1/2}.$$

The problem is: how do we choose w is that we end up with $P(V_j) = q_j$? It was shown by Aurenhammer, Hoffmann, Aronov (1998) that this corresponds to minimizing

$$F(w) = \sum_j \left(q_j w_j - \int_{V_j} [\|x - y_j\|^2 - w_j] dP(x) \right).$$

Merigot (2011) gives a multiscale method to minimize $F(w)$.

There are a few papers (Merigot 2011 and Gerber and Maggioni 2017) use multiscale methods for computing the distance. These approaches make use of decompositions like those used for the minimax theory.

Cuturi (2013) showed that if we replace $\inf \mathbb{E} \|x - y\|^p dJ(x, y)$ with the regularized version $\inf \mathbb{E} \|x - y\|^p dJ(x, y) + \int j(x, y) \log j(x, y)$ then a minimizer can be found using a fast, iterative algorithm called the Sinkhorn algorithm. However, this requires discretizing the space and it changes the metric.

Finally, recall that, if $P = N(\mu_1, \Sigma_1)$ and $Q = N(\mu_2, \Sigma_2)$ then

$$W^2(P, Q) = \|\mu_1 - \mu_2\|^2 + B^2(\Sigma_1, \Sigma_2)$$

where

$$B^2(\Sigma_1, \Sigma_2) = \text{tr}(\Sigma_1) + \text{tr}(\Sigma_2) - 2\text{tr} \left[(\Sigma_1^{1/2} \Sigma_2 \Sigma_1^{1/2})^{1/2} \right].$$

Clearly computing the distance is easy in this case.

10 Applications

The Wasserstein distance is now being used for many tasks in statistical machine learning including:

- Two-sample testing without smoothness
- goodness-of-fit
- analysis of mixture models
- image processing
- dimension reduction
- generative adversarial networks
- domain adaptation
- signal processing

The domain adaptation application is very intriguing. Suppose we have two data sets $\mathcal{D}_1 = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ and $\mathcal{D}_2 = \{(X'_1, Y'_1), \dots, (X'_N, Y'_N)\}$ from two related problems. We want to construct a predictor for the first problem. We could use just \mathcal{D}_1 . But if we can find a transport map T that makes \mathcal{D}_2 similar to \mathcal{D}_1 , then we can apply the map to \mathcal{D}_2 and effectively increase the sample size for problem 1. This kind of reasoning can be used for many statistical tasks.

11 Summary

Wasserstein distance has many nice properties and has become popular in statistics and machine learning. Recently, for example, it has been used for Generative Adversarial Networks (GANs).

But the distance does have problems. First, it is hard to compute. Second, as we have seen, we do not have a way to do inference for the distance. This reflects the fact that the distance is not a smooth functional which is, itself not a good thing. We have also seen that the distance is not robust although, the trimmed version may fix this.

High-Dimensional, Two-Sample Testing

1 Introduction

We observe two iid sample

$$X_1, \dots, X_n \sim P, \quad Y_1, \dots, Y_m \sim Q$$

where $X_i, Y_i \in \mathbb{R}^d$. We want to test

$$H_0 : P = Q \quad \text{versus} \quad H_1 : P \neq Q.$$

Throughout, we will assume that $n/(n+m) \rightarrow \pi \in (0, 1)$ as the sample size increases.

In low dimensions, there are many tests with good power. For example, we could use the test statistic

$$T = \sup_t |\widehat{F}_n(t) - \widehat{G}_n(t)|$$

where \widehat{F}_n and \widehat{G}_n are the empirical cdf's. To find the α -level critical value we can use asymptotic theory or permutation testing. But there are other approaches for the high-dimensional case.

Why are we interested in two-sample testing? We might be interested in testing whether two groups are the same for scientific reasons (treatment versus control, for example). Two sample testing can also be used to screen features for classification.

2 Metrics

One way to define a test is to first define a metric between distributions. For example

$$d(P, Q) = \sup_{g \in \mathcal{G}} \left| \int g dP - \int g dQ \right|$$

for some class of functions \mathcal{G} . Here are some examples. If $\mathcal{G} = \{g : \|g\|_\infty \leq 1\}$ then $d(P, Q)$ is the total variation distance. If \mathcal{G} is the set of g such that

$$\sup_{x \neq y} \frac{|g(y) - g(x)|}{\|x - y\|} \leq 1$$

then $d(P, Q)$ is the earth-mover distance (or Wasserstein distance). This is equivalent to $\inf_R \mathbb{E}_R \|X - Y\|$ where the infimum is over all joint distributions R for (X, Y) with marginals

P and Q . If $\mathcal{G} = \{I_{(-\infty, t]} : t \in \mathbb{R}^d\}$ then $d(P, Q)$ is the Kolmogorov-Smirnov distance. See Sriperumbudur et al (2010) for more examples.

In general, estimating $d(P, Q)$ is difficult. But if we take \mathcal{G} to be a RKHS defined by a kernel K , it can be shown that

$$\theta = d^2(P, Q) = \int \int K(x, y) dP(x) dP(y) + \int \int K(x, y) dQ(x) dQ(y) - 2 \int \int K(x, y) dP(x) dQ(y).$$

The plus-in estimator of $d^2(P, Q)$ is

$$T = \frac{2}{n(n-1)} \sum_{i < j} K(X_i, X_j) + \frac{2}{m(m-1)} \sum_{i < j} K(Y_i, Y_j) - \frac{2}{nm} \sum_{i,j} K(X_i, Y_j).$$

A related distance is the energy distance (Szekely 1989, 2002) defined by

$$d^2(P, Q) = 2\mathbb{E}[||X - Y||] - \mathbb{E}[||X - X'||] - \mathbb{E}[||Y - Y'||].$$

The advantage of the energy distance is that there is no tuning parameter. (The RKHS distance actually requires a bandwidth.) The sample estimate is

$$\frac{2}{n_1 n_2} \sum_i \sum_j ||X_i - Y_j|| - \frac{1}{n_1^2} \sum_i \sum_j ||X_i - X_j|| - \frac{1}{n_2^2} \sum_i \sum_j ||Y_i - Y_j||.$$

How do we know when to reject H_0 ? One approach is to find the limiting distribution of T under H_0 . This turns out to be, for the RKHS distance,

$$T \rightsquigarrow 2 \sum_{j=1}^{\infty} \lambda_j (Z_j^2 - 1)$$

where the Z_j 's are $N(0,1)$ and the λ_j 's are the eigenvalues defined by

$$\int L(x, y) \psi_j(x) dP(x) = \lambda_j \psi_j(y)$$

where $L(x, y) = K(x, y) - \mathbb{E}[K(x, X)] - \mathbb{E}[K(X, x)] + \mathbb{E}[K(X, Y)]$. This distribution is called a *Gaussian chaos*. This distribution has infinitely many nuisance parameters which makes it un-useable. Instead, we use the permutation distribution to choose the critical value.

It can be shown that

$$T - d^2(P, Q) = O_P\left(\frac{1}{\sqrt{N}}\right)$$

where $N = n \wedge m$. Thus, it appears that the quality of T does not depend on the dimension! This is false. What matters here is the power. As we shall see below, the minimax power, that is the smallest detectable difference, is

$$\left(\frac{1}{N}\right)^{\frac{2\beta}{4\beta+d}}$$

where β is the smoothness. This was proved by Arias-Castro, Pelletier and Saligrama (2016) based on techniques developed by Ingster (1987). We'll discuss this more below.

The problem is that the kernel is hiding a lot. To see this, note that T is essentially the same as

$$\int (\hat{p}_h(x) - \hat{q}_h(x))^2$$

where \hat{p}_h and \hat{q}_h are kernel density estimators. This test was proposed by Anderson, Hall and Titterington (1994). But remember, the kernel has a tuning parameter. If it is Gaussian, there is a bandwidth. The statement $T - d(P, Q) = O_P(1/\sqrt{N})$ assumes we do not change the bandwidth. But to have good power, we need to let the bandwidth go to zero and we no longer have the fast rate. The power of the RKHS test in general, nonparametric settings is not well studied.

Now suppose we want a confidence interval for $\theta = d^2(P, Q)$. Unfortunately, there is no known practical method if we use the above estimator. However, we can use the idea in Gretton et al (2012) to get a simple (but statistically inefficient) method. Instead of using a U -statistic, we break the sample into blocks of size two. For simplicity, assume that $n_1 = n_2 = n$. Define

$$\hat{\theta} = \frac{2}{n} \sum_j h((X_{2j-1}, Y_{2j-1}), (X_{2j}, Y_{2j})) \equiv \frac{1}{m} \sum_j R_j$$

where $m = n/2$ and

$$h((x_i, y_i), (x_j, y_j)) = K(X_i, X_j) + K(Y_i, Y_j) - K(X_i, Y_j) - K(X_j, Y_i).$$

It follows from the CLT and Slutsky's theorem that $\sqrt{m}(\hat{\theta} - \theta)/s \rightsquigarrow N(0, 1)$ where s^2 is the sample variance of R_1, \dots, R_m . Hence, an asymptotic $1 - \alpha$ confidence interval is $\hat{\theta} \pm sz_{\alpha/2}/\sqrt{m}$.

3 Graph Based Tests

Another class of tests is based on geometric graphs. Let Z_1, \dots, Z_N be the combined sample where $N = n + m$. Let $L_i = 1$ if Z_i is from group 1 and $L_i = 2$ if Z_i is from group 2.

Let N_i be the k -nearest neighbors of Z_i . Define

$$T = \frac{1}{nk} \sum_{i=1}^n \sum_{r=1}^k B_j(r)$$

where $B_j(r) = 1$ if the r^{th} nearest neighbor has the same label as Z_i . This corresponds to forming a k nearest neighbor graph and asking how many of the k nearest neighbors are

from the same group as the node. The probability of getting the same label under H_0 is $\mu = \pi^2 + (1 - \pi)^2$.

It can be shown that, under H_0 ,

$$\frac{\sqrt{nk}(T - \mu)}{\sigma} \rightsquigarrow N(0, 1).$$

The proof is difficult because the test statistic is summing quantities that are not independent. The variance σ^2 is known but is very, very complicated. See Schilling (1986a, 1986b). In practice, we can use the permutation distribution to get the critical value. Under H_1 , the mean of T converges to

$$\theta = 1 - 2\pi(1 - \pi) \int \frac{p(x)q(x)}{\pi p(x) + (1 - \pi)q(x)} dx$$

which is a distance between p and q . In my experience, this test works well even with $k = 1$.

In high-dimensions we need to correct the test to account for some strange effects (Mondal, Biswas and Ghosh, 2015). If P concentrates its data on a ring R and Q concentrates its data on a larger ring S that surrounds R , then every point in Q can be closer to a point from P .

Here is an example. Let's take $k = 1$ and $n = m$. Let $B_i = 1$ if its nearest neighbor is from the same group. The test statistic is $T = n^{-1} \sum_i B_i$. We are testing

$$H_0 : P(B_i) = \frac{1}{2} \quad \text{versus} \quad H_1 : P(B_i) > \frac{1}{2}.$$

Suppose that $X_1, X_2 \sim N(\mu_1, \sigma_1^2 I)$ and $Y_1, Y_2 \sim N(\mu_2, \sigma_2^2 I)$. Take $\mu_1 = (a, \dots, a)$ and $\mu_2 = (b, \dots, b)$. Now,

$$\frac{1}{d} \|X_1 - X_2\|^2 \xrightarrow{P} 2\sigma_1^2, \quad \frac{1}{d} \|Y_1 - Y_2\|^2 \xrightarrow{P} 2\sigma_2^2, \quad \frac{1}{d} \|X_1 - Y_2\|^2 \xrightarrow{P} \sigma_1^2 + \sigma_2^2 + (a - b)^2.$$

Let $a = 0$, $b = 0.2$, $\sigma_1^2 = 1$, $\sigma_2^2 = 1.2$. Then

$$2\sigma_1^2 < \sigma_1^2 + \sigma_2^2 + (a - b)^2 < 2\sigma_2^2.$$

Every observation from Q is closer to an observation from P .

The data will look like this:

	X_1	X_2	\dots	X_n	Y_1	Y_2	\dots	Y_n
B_i	1	1	\dots	1	0	0	\dots	0

We will not reject H_0 in this case since $(2n)^{-1} \sum_i B_i = 1/2$. The problem is that $P(B_i = 1 | L_i = 1) = 1$ and $P(B_i = 1 | L_i = 2) = 0$ but $P(B_i = 1) = 1/2$. However, if we do a

two-sided test, separately within each group, we would reject. Mondal, Biswas and Ghosh (2015) suggest taking

$$U = (T_1 - \theta)^2 + (T_2 - \theta)^2$$

where $T_j = (nk)^{-1} \sum_{i:L_i=j} \sum_{Z_j \in N_i} I(L_i = L_j)$. However, this test can have low power in other cases. The best strategy is to use both tests i.e. $W = T \vee U$.

A similar test, called the *cross-match test*, was defined by Rosenbaum (2005). We take the pooled sample and partition the data into pairs $W_1 = (Z_1, Z_2), W_2 = (Z_3, Z_4), \dots$. The partition is chosen to minimize $\sum_j \|Z_{2j} - Z_{2j-1}\|^2$. Let

$$T = \sum_i A_i$$

where $A_i = 1$ if the i^{th} pair has differing labels (i.e. (0,1) or (1,0)) and $A_i = 0$ otherwise. We reject when T is small. The exact distribution of T under H_0 is known; it is hypergeometric. It can accurately be approximated with a $N(\mu, \sigma^2)$ where

$$\mu = \frac{mn}{(N-1)}, \quad \sigma^2 = \frac{2n(n-1)m(m-1)}{(N-3)(N-1)^2}.$$

This accurate, simple limiting distribution for T under the null is the main advantage of this test. However, seems to have less power than the NN test. Also, the distribution of T under H_1 is not known. We could have defined $T = \sum_i B_i$ where $B_i = 1 - A_i$ and rejected when T is large. This is then the same as the k -NN test with $k = 1$ except that we allow no overlap between groups.

4 Smooth Tests

Neyman (1937) introduced a method for testing that takes advantage of smoothness. First, consider one dimensional data $Y_1, \dots, Y_n \sim P$. Suppose we want to test

$$H_0 : P = \text{Uniform}(0, 1) \quad H_1 : P \neq \text{Uniform}(0, 1).$$

If we want to have power against smooth alternatives, Neyman proposed that we define

$$p_\theta(x) = c(\theta) \exp \left(\sum_{j=1}^k \theta_j \psi_j(x) \right)$$

where ψ_1, ψ_2, \dots , are orthonormal functions and

$$c(\theta) = \frac{1}{\int \exp \left(\sum_{j=1}^k \theta_j \psi_j(x) \right) dx}.$$

The null hypothesis corresponds to $\theta = (\theta_1, \dots, \theta_k) = (0, \dots, 0)$. One way to test H_0 is to use the likelihood ratio test $T = 2(\ell(\hat{\theta}) - \ell(0))$. Under H_0 , $T \rightsquigarrow \chi_k^2$. But Neyman pointed out that there is a computationally easier test,

$$U = n \sum_j \bar{\psi}_j^2$$

where

$$\bar{\psi}_j = \frac{1}{n} \sum_i \psi_j(X_i).$$

This also has the property that, under H_0 , $U \rightsquigarrow \chi_k^2$. But it avoids having to deal with the normalizing constant.

Now we move to the two-sample case. Let $F(t) = P(X \leq t)$ and $G(t) = Q(Y \leq t)$. Let $Z = F(Y)$. Then the cdf of Z is

$$H(z) = \mathbb{P}(Z \leq z) = \mathbb{P}(F(Y) \leq z) = \mathbb{P}(Y \leq R(z)) = G(R(z))$$

where $R(z) = F^{-1}(z)$. Under H_0 , $Z \sim \text{Unif}(0, 1)$. Now H has density

$$\rho(z) = \frac{q(F^{-1}(z))}{p(F^{-1}(z))}$$

and $\rho(z) = 1$ under H_0 . Bera, Ghosh and Xiao (2013) suggest using the family

$$\rho_\theta(z) = c(\theta) \exp \left(\sum_{j=1}^k \theta_j \psi_j(x) \right).$$

Their test statistic is $m \bar{\psi}^T \bar{\psi}$ where

$$\bar{\psi}_j = \frac{1}{m} \sum_i \psi_j(V_i)$$

and $V_i = \hat{F}_n(Y_i)$. Bera, Ghosh and Xiao (2013) prove that the statistic again has a limiting χ_k^2 distribution.

Zhou, Zheng and Zhang (arXiv:1509.03459) considered the high-dimensional case. They consider all one-dimensional projections of the data. Their test is

$$T = \sqrt{\frac{nm}{n+m}} \sup_u T(u)$$

where the supremum is over the $d-1$ -dimensional sphere and $T(u)$ is the Bera-Ghosh-Xiao statistic based on the one-dimensional data $u^T Y_i$. They also allow the parameter k to be chosen from the data. (In fact, they maximize the test over k .)

The limiting distribution of T under H_0 is complicated: it is the supremum of a Gaussian process. To get a practical test there are two possibilities. One is to use permutations. The other is based on a version of the bootstrap called *the multiplier bootstrap*. Their simulations suggest that this test works well. But it is unclear how it compares to the other tests.

5 Histogram Test

Under smoothness assumptions and compact support, Ingster (1987) showed that optimal tests can be obtained using histograms. Arias-Castro, Pelletier and Saligrama (2016) extended this to the multivariate case. Assume smoothness level β . For simplicity let $m = n$. Form a histogram with $N \approx n^{2/(4\beta+1)}$ bins. Set

$$T = \sum_j (C_j - D_j)^2$$

where C_j is the number of X_i 's in bin j and let D_j is the number of Y_i 's in bin j . We reject for T large. This test is, in theory, optimal. In fact, Ingster later showed that the test can be made adaptive to the degree of smoothness.

6 Sparsity

Let us write

$$X_i = (X_i(1), \dots, X_i(d)), \quad Y_i = (Y_i(1), \dots, Y_i(d)).$$

In some cases, we might suspect that P and Q only differ in a few features. In other words, there is sparsity. If so, the easiest thing is to do all the one-dimensional marginal tests and a Bonferroni correction. Let T_j be your favorite one dimensional test applied to the j th feature only. The the statistic to be $T = \vee_j T_j$. This test will have good power in the sparse case and it is very easy to compute.

7 Minimax Theory

What does it mean for a test to be optimal? Just as there is a theory for minimax estimation, there is also a theory for minimax testing. We discussed this a few weeks ago. I'll remind you of a few basic facts.

To keep it simple, suppose that $m = n$. We want to test $H_0 : P = Q$. Let \mathcal{P} be a set of distributions and assume that $P, Q \in \mathcal{P}$.

Recall that a level α test is a function ϕ of the data taking values 0 or 1 such that $P(\phi = 1) \leq \alpha$ for every $P \in H_0$. Let Φ_n denote all level α tests. The minimax type II error, for a set of distributions \mathcal{P} is

$$\beta_n(\epsilon) = \inf_{\phi \in \Phi_n} \sup_{P, Q} P^n(\phi = 0)$$

where the supremum is over all $P, Q \in \mathcal{P}$ such that $d(P, Q) > \epsilon$. Fix any small $\delta > 0$. We say that the minimax separation is ϵ_n if $\epsilon < \epsilon_n$ implies that $\beta_n(\epsilon) \geq \delta$.

If \mathcal{P} is the β smoothness class and d is the L_2 distance between densities, then Arias-Castro, Pelletier and Saligrama (2016) show that

$$\epsilon_n \asymp \left(\frac{1}{n}\right)^{\frac{2\beta}{4\beta+d}}.$$

The minimax risk is achieved by the histogram test.

8 Discrete Distributions

Suppose that X_i and Y_i are discrete random variables taking values in $\{1, \dots, d\}$. Let

$$C_j = \#\{X_i = j\}, \quad D_j = \#\{Y_i = j\}.$$

Let $C = (C_1, \dots, C_d)$ and $D = (D_1, \dots, D_d)$. These are multinomial and we can test $H_0 : P = Q$ using a likelihood ratio test or χ^2 test.

But when d is large, the usual tests might have poor power. Improved tests have been developed by Chan et al (2014) and Diakonikolas and Kane (2016). for example. Moreover, these tests are designed to have good power against alternatives with respect to total variation distance. For example, Chan et al propose the test statistic

$$T = \sum_j \frac{(C_j - D_j)^2 - (C_j + D_j)}{C_j + D_j}.$$

We reject when T is large. The prove that this test has good power as long as $\text{TV}(P, Q) > d^{1/4}/\sqrt{n}$ which is the minimax bound.

References

Anderson, Hall and Titterington (1994). Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis*, 41-54.

Arias-Castro, Pelletier and Saligrama (2016). arXiv:1607.08156.

Berlinet and Thomas-Agnan (2011). *Reproducing kernel Hilbert spaces in probability and statistics*, Springer.

Chan, Siu-On, et al. "Optimal algorithms for testing closeness of discrete distributions." Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2014.

Gretton, Borgwardt, Rasch, Malte, Scholkopf and Smola (2007). A kernel method for the two-sample-problem. NIPS.

Henze, N. (1988). A multivariate two-sample test based on the number of nearest neighbor type coincidences. *The Annals of Statistics*, 772-783.

Ingster, Y. (1987) Minimax testing of nonparametric hypotheses on a distribution density in the L_p metrics. *Theory of Probability and Its Applications*, 333-337.

Mondal, Biswas and Ghosh (2015). On high dimensional two-sample tests based on nearest neighbors. *Journal of Multivariate Analysis*, 168-178.

Rosenbaum (2005). An exact distribution-free test comparing two multivariate distributions based on adjacency. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 515-530.

Schilling, Mark F. "Multivariate two-sample tests based on nearest neighbors." *Journal of the American Statistical Association* 81.395 (1986): 799-806.

Schilling, M. F. "Mutual and shared neighbor probabilities: finite-and infinite-dimensional results." *Advances in Applied Probability* 18.02 (1986): 388-405.

Sriperumbudur, Bharath K., et al. "Hilbert space embeddings and metrics on probability measures." *Journal of Machine Learning Research* 11.Apr (2010): 1517-1561.

Szkeley and Rizzo (2004). Testing for equal distributions in high dimension. *InterStat*, 1-6.

Dimension Reduction and Hidden Structure

We consider two related problems: (i) using low dimensional approximations for dimension reduction and (ii) estimating low dimensional structure.

1 Dimension Reduction

We might not want to estimate lower dimensional structure. Instead, we might just want to use a low dimensional approximation to the data to make other tasks easier. This is dimension reduction.

1.1 Principal Component Analysis (PCA)

Principal components analysis (PCA) finds low dimensional approximations to the data by projecting the data onto linear subspaces.

Let $X \in \mathbb{R}^d$ and let \mathcal{L}_k denote all k -dimensional linear subspaces. The k^{th} principal subspace is

$$\ell_k = \underset{\ell \in \mathcal{L}_k}{\operatorname{argmin}} \mathbb{E} \left(\min_{y \in \ell} \|\tilde{X} - y\|^2 \right)$$

where $\tilde{X} = X - \mu$ and $\mu = \mathbb{E}(X)$. The dimension-reduced version of X is then $T_k(X) = \mu + \pi_{\ell_k} X$ where $\pi_{\ell_k} X$ is the projection of X onto ℓ_k . To find ℓ_k proceed as follows.

Let $\Sigma = \mathbb{E}((X - \mu)(X - \mu)^T)$ denote the covariance matrix, where $\mu = \mathbb{E}(X)$. Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$ be the ordered eigenvalues of Σ and let e_1, \dots, e_d be the corresponding eigenvectors. Let Λ be the diagonal matrix with $\Lambda_{jj} = \lambda_j$ and let $E = [e_1 \cdots e_d]$. Then the spectral decomposition of Σ is

$$\Sigma = E \Lambda E^T = \sum_j \lambda_j e_j e_j^T.$$

Theorem 1 *The k^{th} principal subspace ℓ_k is the subspace spanned by e_1, \dots, e_k . Furthermore,*

$$T_k(X) = \mu + \sum_{j=1}^k \beta_j e_j$$

where $\beta_j = \langle X - \mu, e_j \rangle$. The risk satisfies

$$R(k) = \mathbb{E} \|X - T_k(X)\|^2 = \sum_{j=k+1}^d \lambda_j.$$

We can restate the result as follows. To minimize

$$\mathbb{E}\|Y_i - \alpha - A\beta_i\|^2,$$

with respect to $\alpha \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times k}$ and $\beta_i \in \mathbb{R}^k$ we set $\alpha = \mu$ and $A = [e_1 \ e_2 \ \dots \ e_k]$. Any other solution is equivalent in the sense that it corresponds to the same subspace.

We can choose k by fixing some α and then taking

$$k = \min \left\{ m : \frac{R(m)}{R(0)} \geq 1 - \alpha \right\} = \min \left\{ m : \frac{\sum_{j=1}^m \lambda_j}{\sum_{j=1}^d \lambda_j} \geq 1 - \alpha \right\}.$$

Let $Y = (Y_1, \dots, Y_d)$ where $Y_i = e_i^T(X - \mu)$. Then Y is the PCA-transformation applied to X . The random variable Y has the following properties:

Lemma 2 *We have:*

1. $\mathbb{E}[Y] = 0$ and $\text{Var}(Y) = \Lambda$.
2. $X = \mu + EY$.
3. $\sum_{j=1}^m \text{Var}(Y_j) = \Sigma_{11} + \dots + \Sigma_{mm}$.

Hence,

$$\frac{\sum_{j=1}^m \lambda_j}{\sum_{j=1}^d \lambda_j}$$

is the percentage of variance explained by the first m principal components.

The data version of PCA is obtained by replacing Σ with the sample covariance matrix

$$\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X}_n)(X_i - \bar{X}_n)^T.$$

Principal Components Analysis (PCA)

1. Compute the sample covariance matrix $\widehat{\Sigma} = n^{-1} \sum_{i=1}^n (X_i - \bar{X}_n)(X_i - \bar{X}_n)^T$.
2. Compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots$ and eigenvectors e_1, e_2, \dots , of $\widehat{\Sigma}$.
3. Choose a dimension k .
4. Define the dimension reduced data $Z_i = T_k(X_i) = \bar{X} + \sum_{j=1}^k \beta_{ij} e_j$ where $\beta_{ij} = \langle X_i - \bar{X}, e_j \rangle$.

Example 3 *Figure 1 shows a synthetic two-dimensional data set together with the first principal component.*

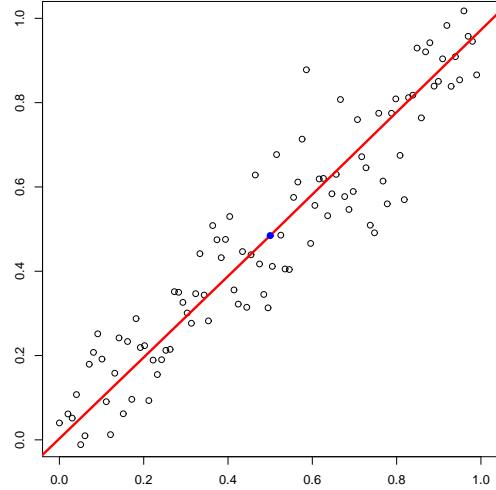


Figure 1: First principal component in a simple two dimensional example.

Example 4 *Figure 2 shows some handwritten digits. The eigenvalues and the first few eigenfunctions are shown in Figures 3 and 4. A few digits and their low-dimensional reconstructions are shown in Figure 5.*

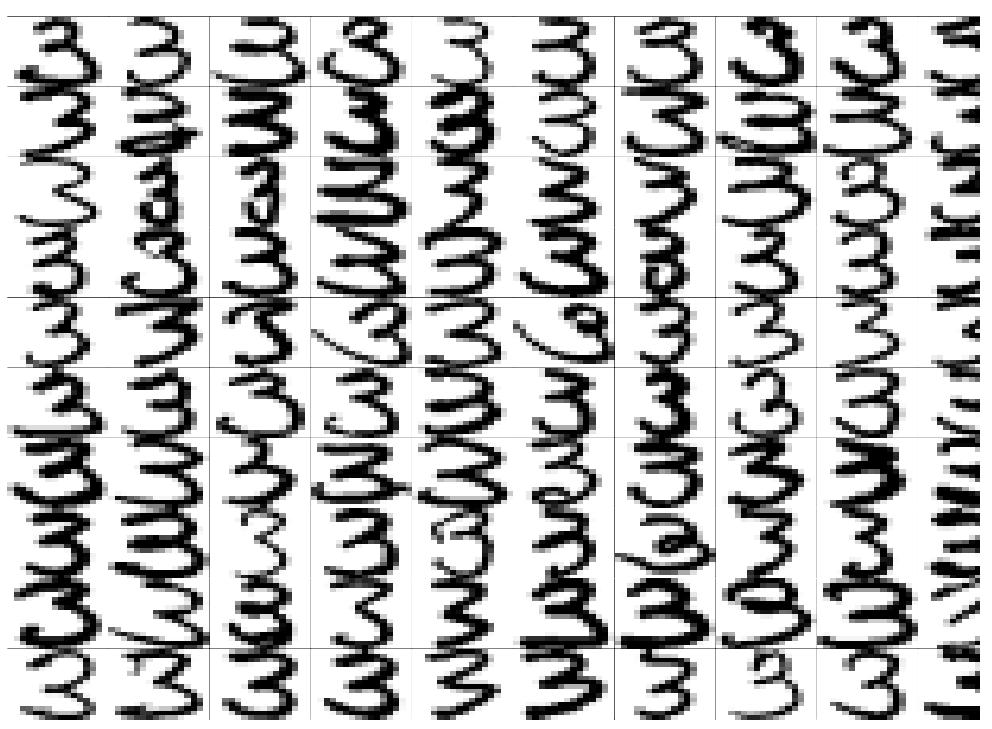


Figure 2: Handwritten digits.

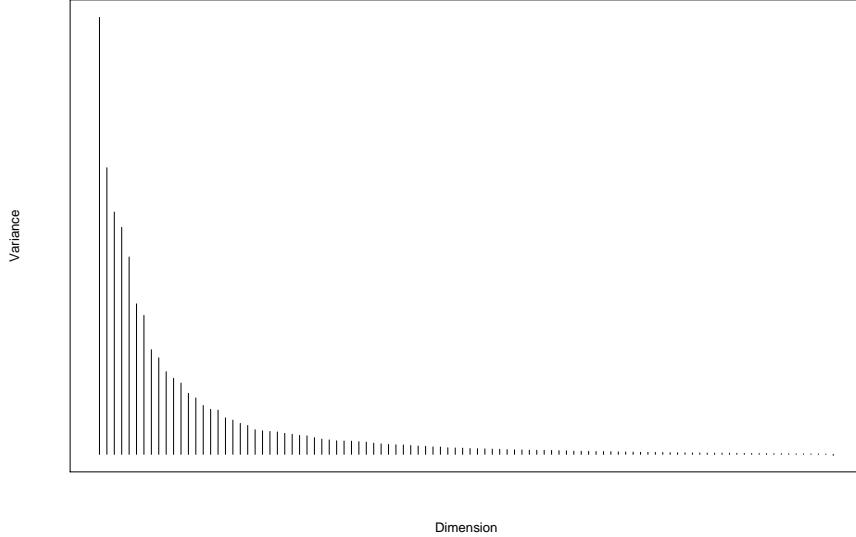


Figure 3: Digits data: eigenvalues

How well does the sample version approximate the population version? For now, assume the dimensions d is fixed and that n is large. We will study the high-dimensional case later where we will use some random matrix theory.

Define the operator norm

$$\|\Sigma\| = \sup \left\{ \frac{\|\Sigma v\|}{\|v\|} : v \neq 0 \right\}.$$

It can be shown that $\|\widehat{\Sigma} - \Sigma\| = O_P(1/\sqrt{n})$. According to Weyl's theorem

$$\max_j |\lambda_j(\widehat{\Sigma}) - \lambda_j(\Sigma)| \leq \|\widehat{\Sigma} - \Sigma\|$$

and hence, the estimated eigenvalues are consistent. We can also say that the eigenvectors are consistent. We have

$$\|\widehat{e}_j - e_j\| \leq \frac{2^{3/2} \|\widehat{\Sigma} - \Sigma\|}{\min(\lambda_{j-1} - \lambda_j, \lambda_j - \lambda_{j+1})}.$$

(See Yu, Wang and Samworth, arXiv:1405.0680.) There is also a central limit theorem for the eigenvalues and eigenvectors which also leads to a proof that the bootstrap is valid. However, these limiting results depend on the distinctness of the eigenvalues.

There is a strong connection between PCA and the singular value decomposition (SVD). Let X be an $n \times d$ matrix. The SVD is

$$X = UDV^T$$

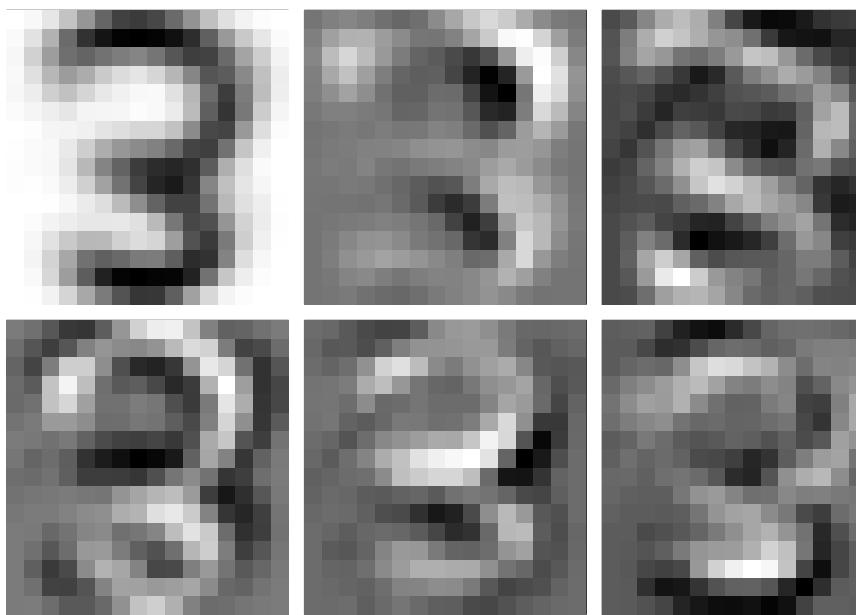


Figure 4: Digits: mean and eigenvectors

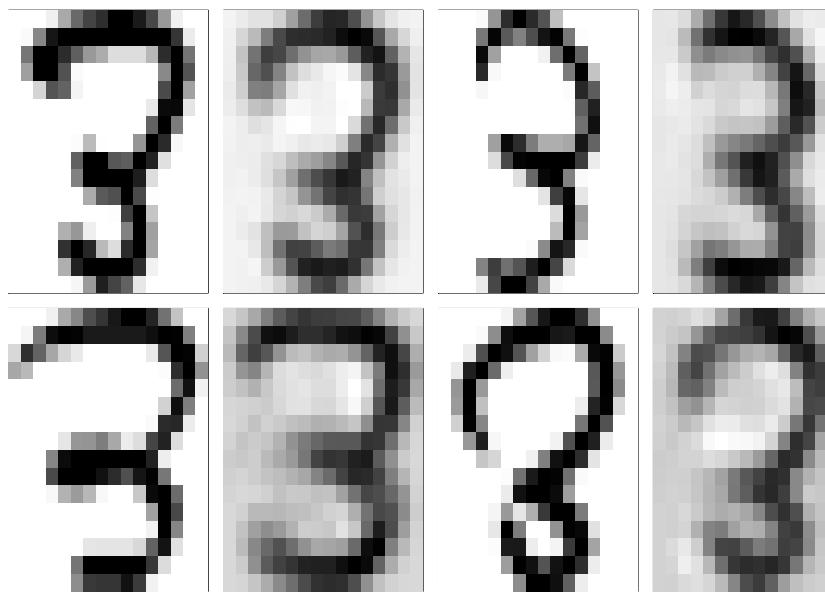


Figure 5: Digits data: Top: digits. Bottom: their reconstructions.

where U is an $n \times n$ matrix with orthonormal columns, V is a $d \times d$ matrix with orthonormal columns, and D is an $n \times d$ diagonal matrix with non-negative real numbers on the diagonal (called singular values). Then

$$X^T X = (V D U^T)(U D V^T) = V D^2 V^T$$

and hence the singular values are the square root of the eigenvalues.

1.2 Multidimensional Scaling

A different view of dimension reduction is provided by thinking in terms of preserving pairwise distances. Suppose that $Z_i = T(X_i)$ for $i = 1, \dots, n$ where $T : \mathbb{R}^d \rightarrow \mathbb{R}^k$ with $k < d$. Define the loss function

$$L = \sum_{i,j} (||X_i - X_j||^2 - ||Z_i - Z_j||^2)$$

which measures how well the map T preserves pairwise distances. **Multidimensional scaling** find the linear map T to minimize L .

Theorem 5 *The linear map $T : \mathbb{R}^d \rightarrow \mathbb{R}^k$ that minimizes L is the projection onto $\text{Span}\{e_1, \dots, e_k\}$ where e_1, \dots, e_k are the first k principal components.*

We could use other measures of distortion. In that case, the MDS solution and the PCA solution will not coincide.

1.3 Kernel PCA

To get a nonlinear version of PCA, we can use a kernel. Suppose we have a “feature map” $x \mapsto \Phi(x)$ and want to carry out PCA in this new feature space. For the moment, assume that the feature vectors are centered (we return to this point shortly). Define the empirical covariance matrix

$$C_\Phi = \frac{1}{n} \sum_{i=1}^n \Phi(x_i) \Phi(x_i)^T.$$

We can define eigenvalues $\lambda_1, \lambda_2, \dots$ and eigenvectors v_1, v_2, \dots of this matrix.

It turns out that the eigenvectors are linear combinations of the feature vectors $\Phi(x_1), \dots, \Phi(x_n)$. To see this, note that

$$\begin{aligned} \lambda v &= C_\Phi v = \frac{1}{n} \sum_{i=1}^n \Phi(x_i) \Phi(x_i)^T v \\ &= \frac{1}{n} \sum_{i=1}^n \langle \Phi(x_i), v \rangle \Phi(x_i) = \sum_{i=1}^n \alpha_i \Phi(x_i) \end{aligned}$$

where

$$\alpha_i = \frac{1}{n} \langle \Phi(x_i), v \rangle = \frac{1}{n\lambda} \langle \Phi(x_i), C_\Phi v \rangle.$$

Now

$$\begin{aligned} \lambda \sum_{i=1}^n \alpha_i \langle \Phi(x_k), \Phi(x_i) \rangle &= \lambda \langle \Phi(x_k), Cv \rangle \\ &= \lambda \langle \Phi(x_k), \frac{1}{n} \sum_{j=1}^n \Phi(x_j) \Phi(x_j)^T v \rangle \\ &= \langle \Phi(x_k), \frac{1}{n} \sum_{j=1}^n \Phi(x_j) \Phi(x_j)^T \sum_{i=1}^n \alpha_i \Phi(x_i) \rangle \\ &= \frac{1}{n} \sum_{i=1}^n \alpha_i \langle \Phi(x_k), \sum_{j=1}^n \langle \Phi(x_j), \Phi(x_i) \rangle \Phi(x_j) \rangle. \end{aligned}$$

Define the kernel matrix K by $K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle$. Then we can write the above equation as

$$\lambda n K \alpha = K^2 \alpha$$

Thus, we need to solve the kernel eigenvalue problem

$$K \alpha = n \lambda \alpha$$

which requires diagonalizing only an $n \times n$ system. Normalizing the eigenvectors, $\langle v, v \rangle = 1$ leads to the condition $\lambda \langle \alpha, \alpha \rangle = 1$. Of course, we need to find all the solutions giving eigenvectors v_1, v_2, \dots .

In order to compute the kernel PCA projection of a new test point x , it is necessary to project the feature vector $\Phi(x)$ onto the principal direction v_m . This requires the evaluation

$$\begin{aligned} \langle v, \Phi(x) \rangle &= \sum_{i=1}^n \alpha_i \langle \Phi(x_i), \Phi(x) \rangle \\ &= \sum_{i=1}^n \alpha_i K(x_i, x) \end{aligned}$$

Thus, the entire procedure uses only the kernel evaluations $K(x, x_i)$ and never requires actual manipulation of feature vectors, which could be infinite dimensional. This is an instance of the *kernel trick*. An arbitrary data point x can then be approximated by projecting $\Phi(x)$ onto the first k vectors. This defines an approximation in the feature space. We then need to find \tilde{x} that corresponds to this projection. There is an iterative algorithm for doing this (Mika et al 1998) which turns out to be a weighed version of the mean shift algorithm.

To complete the description of the algorithm, it is necessary to explain how to center the data in feature space using only kernel operations. This is accomplished by transforming the kernel according to

$$\tilde{K}_{ij} = (K - \mathbf{1}_n K - K \mathbf{1}_n + \mathbf{1}_n K \mathbf{1}_n)_{ij}$$

where

$$\mathbf{1}_n = \frac{1}{n} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} = \frac{1}{n} \mathbf{1} \mathbf{1}^T$$

where $\mathbf{1}$ denotes the vector of all ones.

Kernel PCA. Given a Mercer kernel K and data X_1, X_2, \dots, X_n

1. Center the kernel
2. Compute $K_{ij} = K(X_i, X_j)$
3. Diagonalize K
4. Normalize eigenvectors $\alpha^{(m)}$ so that $\langle \alpha^{(m)}, \alpha^{(m)} \rangle = \frac{1}{\lambda_m}$
5. Compute the projection of a test point x onto an eigenvector v_m by

$$\langle v_m, \Phi(x) \rangle = \sum_{i=1}^n \alpha_i^{(m)} K(X_i, x)$$

Just as for standard PCA, this selects components of high variance, but in the feature space of the kernel. In addition, the “feature functions”

$$f_m(x) = \sum_{i=1}^n \alpha_i^{(m)} K(X_i, x)$$

are orthogonal and act as representative feature functions in the reproducing kernel Hilbert space of the kernel, with respect to the given data. Intuitively, these functions are smooth with respect to the RKHS norm $\|\cdot\|_K$ among all those supported on the data.

Another perspective on kernel PCA is that it is doing MDS on the kernel distances $d_{ij} = \sqrt{2(1 - K(X_i, X_j))}$; see Williams (2002).

1.4 Local Linear Embedding

Local Linear Embedding (LLE) (Roweis et al) is another nonlinear dimension reduction method. The LLE algorithm is comprised of three steps. First, nearest neighbors are computed for each point $X_i \in \mathbb{R}^d$. Second, each point is regressed onto its neighbors, giving weights w_{ij} so that $X_i = \sum_j w_{ij} X_j$. Third, the $X_i \in \mathbb{R}^d$ are replaced by $Y_i \in \mathbb{R}^m$ where typically $m \ll d$ by solving a sparse eigenvector problem. The result is a highly nonlinear embedding, but one that is carried out by optimizations that are not prone to local minima. Underlying the procedure, as for many “manifold” methods, is a weighted sparse graph that represents the data.

Step 1: Nearest Neighbors. Here the set of the K nearest neighbors in standard Euclidean space is constructed for each data point. Using brute-force search, this requires $O(n^2d)$ operations; more efficient algorithms are possible, in particular if *approximate* nearest neighbors are calculated. The number of neighbors K is a parameter to the algorithm, but this is the only parameter needed by LLE.

Step 2: Local weights. In this step, the local geometry of each point is characterized by a set of weights w_{ij} . The weights are computed by reconstructing each input X_i as a linear combination of its neighbors, as tabulated in Step 1. This is done by solving the least squares problem

$$\min_w \sum_{i=1}^n \|X_i - \sum_j w_{ij} X_j\|_2^2 \quad (1)$$

The weights w_{ij} are constrained so that $w_{ij} = 0$ if X_j is not one of the K nearest neighbors of X_i . Moreover, the weights are normalized to sum to one: $\sum_j w_{ij} = 1$, for $i = 1, \dots, n$. This normalization ensures that the optimal weights are invariant to rotation, translation, and scaling.

Step 3: Linearization. In this step the points $X_i \in \mathbb{R}^d$ are mapped to $Y_i \in \mathbb{R}^m$, where m selected by the user, or estimated directly from the data. The vectors Y_i are chosen to minimize the reconstruction error under the local linear mappings constructed in the previous step. That is, the goal is to optimize the functional

$$\Psi(y) = \sum_{i=1}^n \|Y_i - \sum_j w_{ij} Y_j\|_2^2 \quad (2)$$

where the weights w_{ij} are calculated in Step 2. To obtain a unique solution, the vectors are “centered” to have mean zero and unit covariance:

$$\sum_i Y_i = 0 \quad \frac{1}{n} \sum_i Y_i Y_i^T = I_m \quad (3)$$

Carrying out this optimization is equivalent to finding eigenvectors by minimizing the quadratic

form

$$\Psi(y) = y^T G y \quad (4)$$

$$= y^T (I - W)^T (I - W) y \quad (5)$$

$$(6)$$

corresponding to a Rayleigh quotient. Each minimization gives one of the lower ($m + 1$) eigenvectors of the $n \times n$ matrix $G = (I - W)^T (I - W)$. The lowest eigenvector has eigenvalue 0, and consists of the all ones vector $(1, 1 \dots, 1)^T$.

Locally Linear Embedding (LLE). Given n data vectors $X_i \in \mathbb{R}^d$,

1. Compute K nearest neighbors for each point;
2. Compute local reconstruction weights w_{ij} by minimizing

$$\Phi(w) = \sum_{i=1}^n \|X_i - \sum_j w_{ij} X_j\|^2 \quad (7)$$

$$\text{subject to} \quad \sum_j w_{ij} = 1; \quad (8)$$

3. Compute outputs $Y_i \in \mathbb{R}^m$ by computing the first m eigenvectors with nonzero eigenvalues for the $n \times n$ matrix $G = (I - W)^T (I - W)$. The reduced data matrix is $[u_1 \dots u_m]$ where u_j are the eigenvectors corresponding to the first (smallest) nonzero eigenvalues of G .

Note that the last step assumes that the underlying graph encoded by the nearest neighbor graph is connected. Otherwise, there may be more than one eigenvector with eigenvalue zero. If the graph is disconnected, then the LLE algorithm can be run separately on each connected component. However, the recommended procedure is to choose K so that the graph is connected.

Using the simplest algorithms, the first step has time complexity $O(dn^2)$, the second step requires $O(nK^3)$ operations, and the third step, using routines for computing eigenvalues for sparse matrices, requires $O(mn^2)$ operations (and $O(n^3)$ operations in the worse case if sparsity is not exploited and the full spectrum is computed). Thus, for high dimensional problems, the first step is the most expensive. Since the third step computes eigenvectors, it shares the property with PCA that as more dimensions are added to the embedding, the previously computed coordinates do not change.

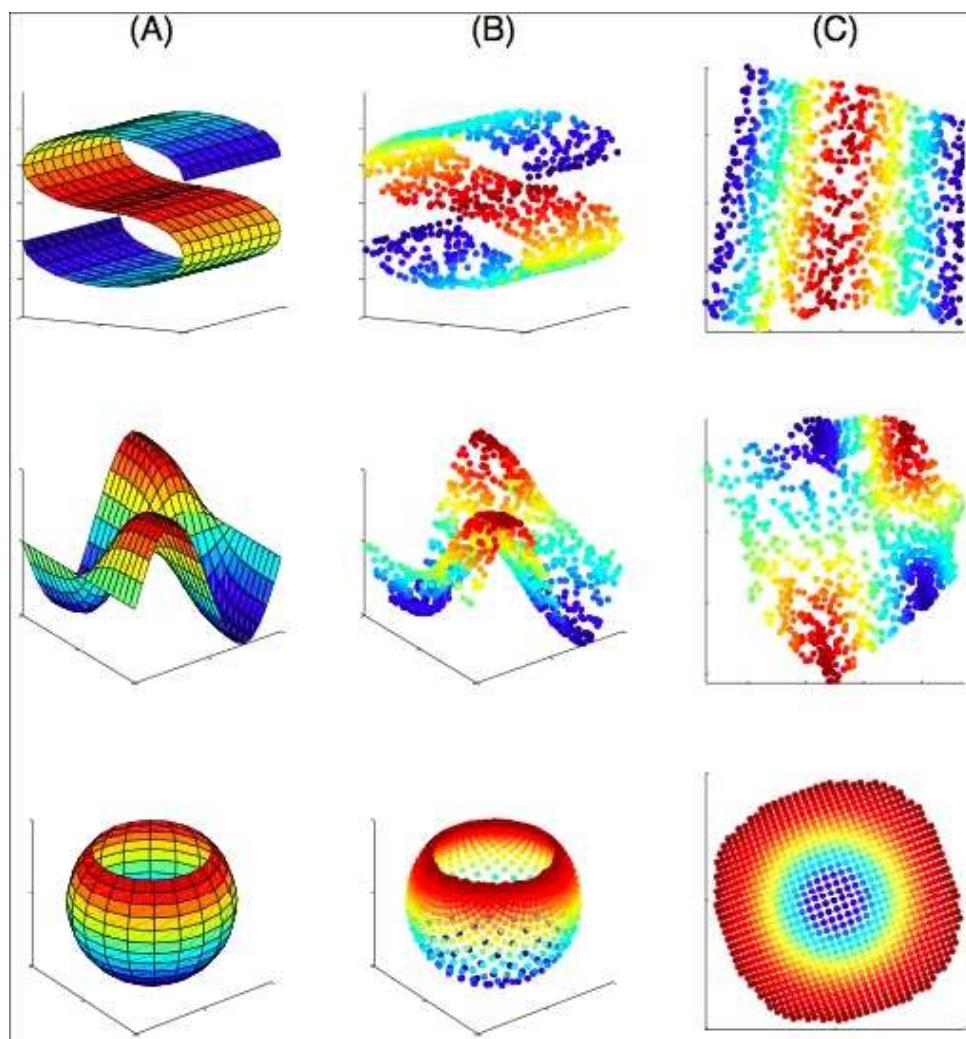


Figure 6: Each data set has $n = 1,000$ points in $d = 3$ dimensions, and LLE was run with $K = 8$ neighbors.

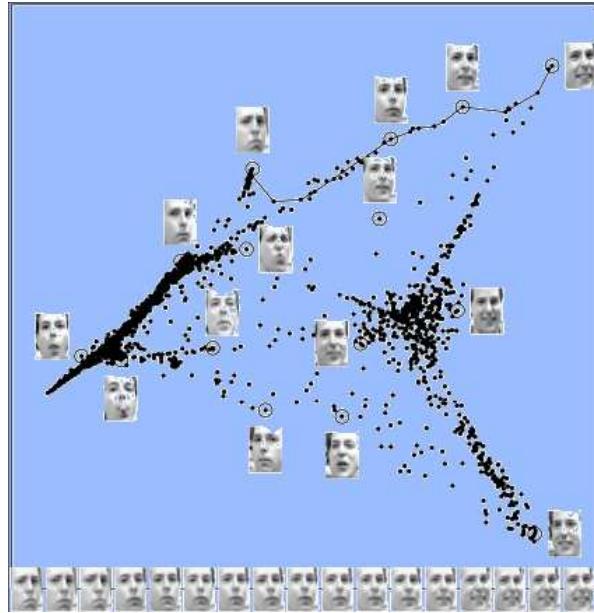


Figure 7: Faces example. $n = 1,965$ images with $d = 560$, with LLE run for $K = 12$ neighbors.

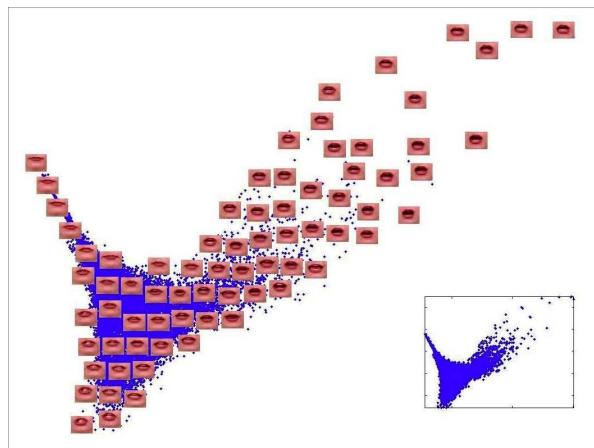


Figure 8: Lips example. $n = 15,960$ images in $d = 65,664$ dimensions, with LLE run for $K = 24$ neighbors.

1.5 Isomap

Isomap is a technique that is similar to LLE, intended to provide a low dimensional “manifold” representation of a high dimensional data set. Isomap differs in how it assesses similarity between objects, and in how the low dimensional mapping is constructed.

The first step in Isomap is to construct a graph with the nodes representing instances $X_i \in \mathbb{R}^d$ to be embedded in a low dimensional space. Standard choices are a k -nearest neighbors, and ϵ -neighborhoods. In the k -nearest neighborhood graph, each point X_i is connected to its closest k neighbors $\mathcal{N}_k(X_i)$, where distance is measured using Euclidean distance in the ambient space \mathbb{R}^d . In the ϵ -neighborhood graph, each point X_i is connected to all points $N_\epsilon(X_i)$ within a Euclidean ball of radius ϵ centered at X_i . The graph $G = (V, E)$ by taking edge set $V = \{x_1, \dots, x_n\}$ and edge set

$$(u, v) \in E \text{ if } v \in \mathcal{N}(u) \text{ or } u \in \mathcal{N}(v) \quad (9)$$

Note that the node degree in these graphs may be highly variable. For simplicity, assume that the graph is connected; the parameters k or ϵ may need to be carefully selected for this to be the case.

The next step is to form a distance between points by taking path distance in the graph. That is $d(X_i, X_j)$ is the shortest path between node X_i and X_j . This distance can be computed for sparse graphs in time $O(|E| + |V| \log |V|)$. The final step is to embed the points into a low dimensional space using metric multi-dimensional scaling.

Isomap. Given n data vectors $X_i \in \mathbb{R}^d$,

1. Compute k nearest neighbors for each point, forming the nearest neighbor graph $G = (V, E)$ with vertices $\{X_i\}$.
2. Compute graph distances $d(X_i, X_j)$ using Dijkstra’s algorithm
3. Embed the points into low dimensions using metric multidimensional scaling

Isomap and LLE both obtain nonlinear dimensionality reduction by mapping points into a low dimensional space, in a manner that preserves the local geometry. This local geometry will *not* be preserved by classical PCA or MDS, since far away points on the manifold will be, typically, be mapped to nearby points in the lower dimensional space.

1.6 Laplacian Eigenmaps

A similar approach is based on the use of the graph Laplacian. Recall that if $w_{ij} = K_h \left(\frac{|X_i - X_j|}{h} \right)$ is a weighting between pairs of points determined by a kernel K , the graph Laplacian associated W is given by

$$L = D - W \quad (10)$$

where $D = \text{diag}(d_i)$ with $d_i = \sum_j w_{ij}$ the sum of the weights for edges emanating from node i . In Laplacian eigenmaps, the embedding is obtained using the spectral decomposition of L .

In particular, let $y_0, y_1, \dots, y_k \in \mathbb{R}^n$ denote the first k eigenvectors corresponding to eigenvalues $0 = \lambda_0 < \lambda_1 < \lambda_2 < \dots < \lambda_{k+1}$ of the Laplacian. This determines an embedding

$$X_i \mapsto (y_{1i}, y_{2i}, \dots, y_{ki}) \in \mathbb{R}^k \quad (11)$$

into $k - 1$ dimensions.

The intuition behind this approach can be seen from the basic properties of Rayleigh quotients and Laplacians. In particular, we have that the first nonzero eigenvector satisfies

$$y_1 = \arg \min y_1^T L y_1 = \arg \min \sum_{i,j} w_{ij} (y_{1i} - y_{1j})^2 \quad (12)$$

$$\text{such that } y_1^T D y_1 = 1 \quad (13)$$

Thus, the eigenvector minimizes the weighted graph L^2 norm; the intuition is that the vector changes very slowly with respect to the intrinsic geometry of the graph. This analogy is strengthened by consistency properties of the graph Laplacian. In particular, if the data lie on a Riemannian manifold M , and $f : M \rightarrow \mathbb{R}$ is a function on the manifold,

$$f^T L f \approx \int_M \|\nabla f(x)\|^2 d_M(x) \quad (14)$$

where on the left hand side we have evaluated the function on n points sampled uniformly from the manifold.

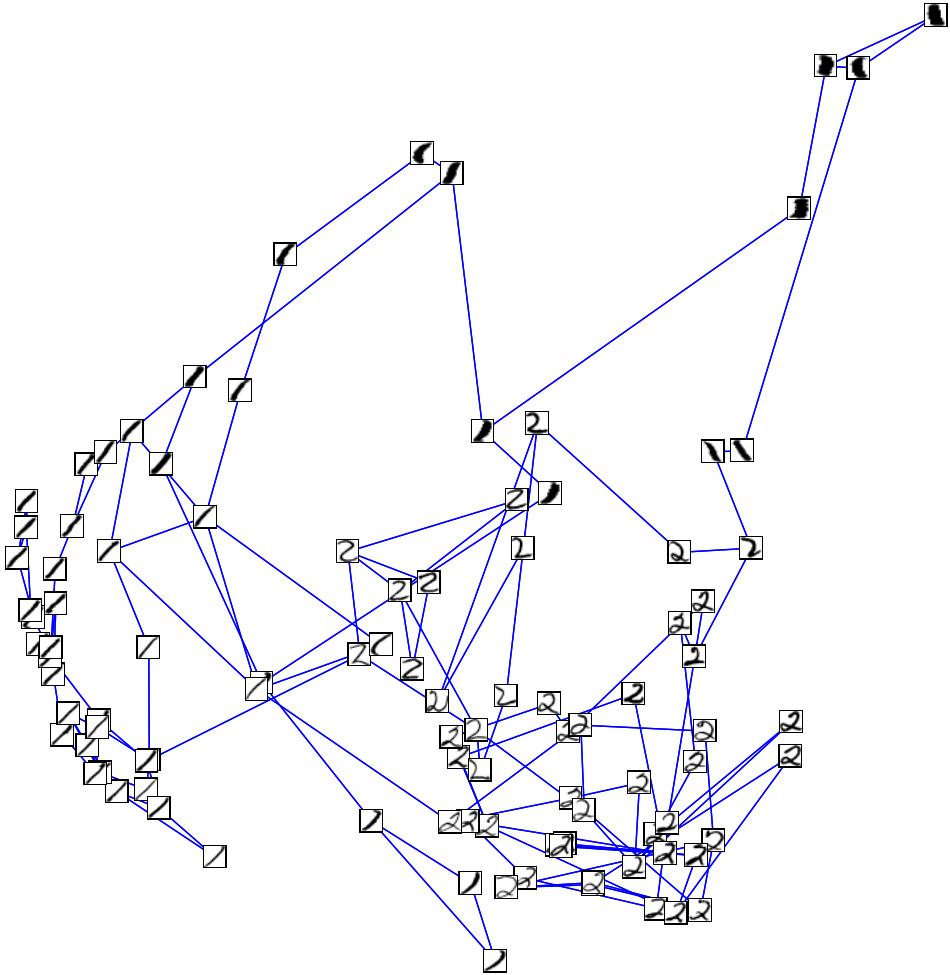


Figure 9: A portion of the similarity graph for actual scanned digits (1s and 2s), projected to two dimensions using Laplacian eigenmaps. Each image is a point in \mathbb{R}^{256} , as a 16×16 pixel image; the graph suggests the data has lower dimensional “manifold” structure

1.7 Diffusion Distances

As we saw when we discussed spectral clustering, there are other versions of graph Laplacians such as $D^{-1/2}WD^{-1/2}$ and $D^{-1}W$ that can have better behavior. In fact, let us consider the matrix $L = D^{-1}W$ which, as we shall now see, has a nice interpretation. We can view L as the transition matrix for a Markov chain on the data. This has a population analogue: we define the diffusion (continuous Markov chain) with transition density

$$\ell(y|x) = \frac{K(x,y)}{s(x)}$$

where $s(x) = \int K(x,y)dP(y)$. The stationary distribution has density $\pi(y) = s(y)/\int s(u)dP(u)$. Then L is just the discrete version of this transition probability. Suppose we run the chain for t steps. The transition matrix is L^t . The properties of this matrix give information on the larger scale structure of the data. We define the *diffusion distance* by

$$D_t(x,y) = \int (q_t(u|x) - q_t(u|y))^2 \frac{p(u)}{\pi(u)}$$

which is a measure of how far it is to get from x to y in t steps (Coifman and Lafon, 2006). It can be shown that

$$D_t(x,y) = \sqrt{\sum_j \lambda_j^{2t} (\psi_j(x) - \psi_j(y))^2}$$

where λ_j and ψ_j are the eigenvalues and eigenvectors of q . We can now reduce the dimension of the data by applying MDS to $D_t(x,y)$. Alternatively, they suggest mapping a point x to

$$\Psi_t(x) = (\lambda_1^t \psi_1(x), \dots, \lambda_k^t \psi_k(x))$$

for some k . An example is shown in Figure 10.

1.8 Principal Curves and Manifolds

A nonparametric generalization of principal components is **principal manifolds**. The idea is to replace linear subspaces with more general manifolds. There are many approaches. We will consider an approach due to Smola et al (2001). However, I should point out that I think ridge estimation is a better way to do this.

Let $X \in \mathbb{R}^d$ and let \mathcal{F} be a set of functions from $[0,1]^k$ to \mathbb{R}^d . The principal manifold (or principal curve) is the function $f \in \mathcal{F}$ that minimizes

$$R(f) = \mathbb{E} \left(\min_{z \in [0,1]^k} \|X - f(z)\|^2 \right). \quad (15)$$

To see how general this is, note that we recover principal components as a special case by taking \mathcal{F} to be linear mappings. We recover k -means by taking \mathcal{F} to be all mappings from

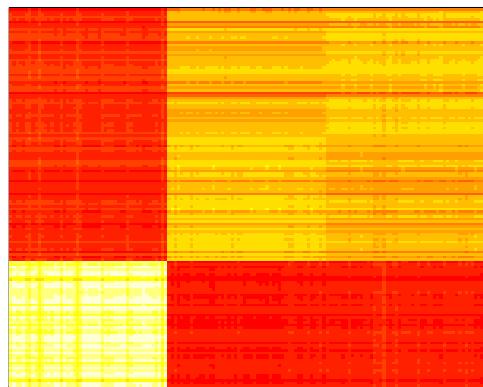
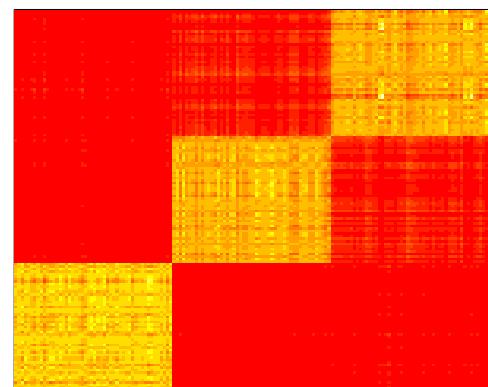
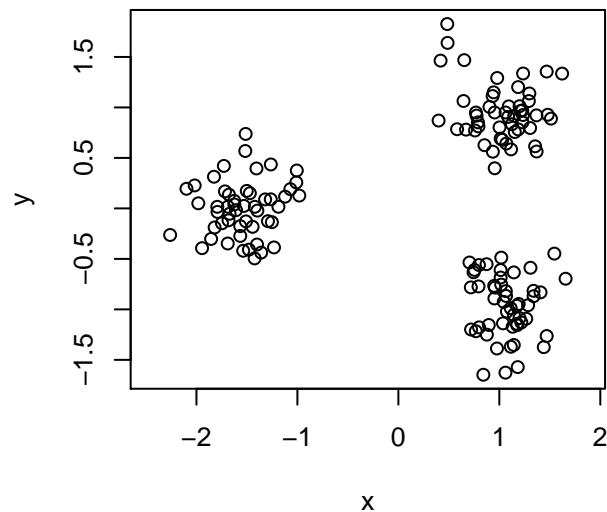


Figure 10: Diffusion maps. Top left: data. Top right: Transition matrix for $t = 1$. Bottom left: Transition matrix for $t = 3$. Bottom right: Transition matrix for $t = 64$.

$\{1, \dots, k\}$ to \mathbb{R}^d . In fact we could construct \mathcal{F} to map to k -lines (or k -planes), also called local principal components; see Bradley and Mangasarian (1998) and Kambhatla and Leen (1994, 1997). But our focus in this section is on smooth curves.

We will take

$$\mathcal{F} = \left\{ f : \|f\|_k^2 \leq C^2 \right\}$$

where $\|f\|_K$ is the norm for a reproducing kernel Hilbert space (RKHS) with kernel K . A common choice is the Gaussian kernel

$$K(z, u) = \exp \left\{ -\frac{\|z - u\|^2}{2h^2} \right\}.$$

To approximate the minimizer, we can proceed as in (Smola, Mika, Schölkopf, Williamson 2001). Fix a large number of points z_1, \dots, z_M and approximate an arbitrary $f \in \mathcal{F}$ as

$$f(z) = \sum_{j=1}^M \alpha_j K(z_j, z)$$

which depends on parameters $\alpha = (\alpha_1, \dots, \alpha_M)$. The minimizer can be found as follows. Define latent variables $\xi = (\xi_1, \dots, \xi_n)$ where $\xi_i \in \mathbb{R}^d$ and

$$\xi_i = \operatorname{argmin}_{\xi \in [0,1]^d} \|X_i - f(\xi)\|^2.$$

For fixed α we find each ξ_i by any standard nonlinear function minimizer. Given ξ we then find α by minimizing

$$\frac{1}{n} \sum_{i=1}^n \|X_i - \sum_{j=1}^M \alpha_j K(z_j, \xi_i)\|^2 + \frac{\lambda}{2} \sum_{i=1}^M \sum_{j=1}^M \alpha_i \alpha_j K(z_i, z_j).$$

The minimizer is

$$\alpha = \left(\frac{\lambda n}{2} K_z + K_\xi^T K_\xi \right)^{-1} K_\xi^T X$$

where $(K_z)_{ij} = K(z_i, z_j)$ is $M \times M$ and $(K_\xi)_{ij} = K(\xi_i, z_j)$ is $n \times M$. Now we iterate, alternately solving for ξ and α .

Example 6 Figure 11 shows some data and four principal curves based on increasing degrees of regularization.

Theoretical results are due to Kégl et al. (2000) and Smola, Mika, Schölkopf, Williamson (2001). For example, we may proceed as follows. Define a norm

$$\|f\|_\# \equiv \sup_{z \in [0,1]^k} \|f(z)\|_2.$$

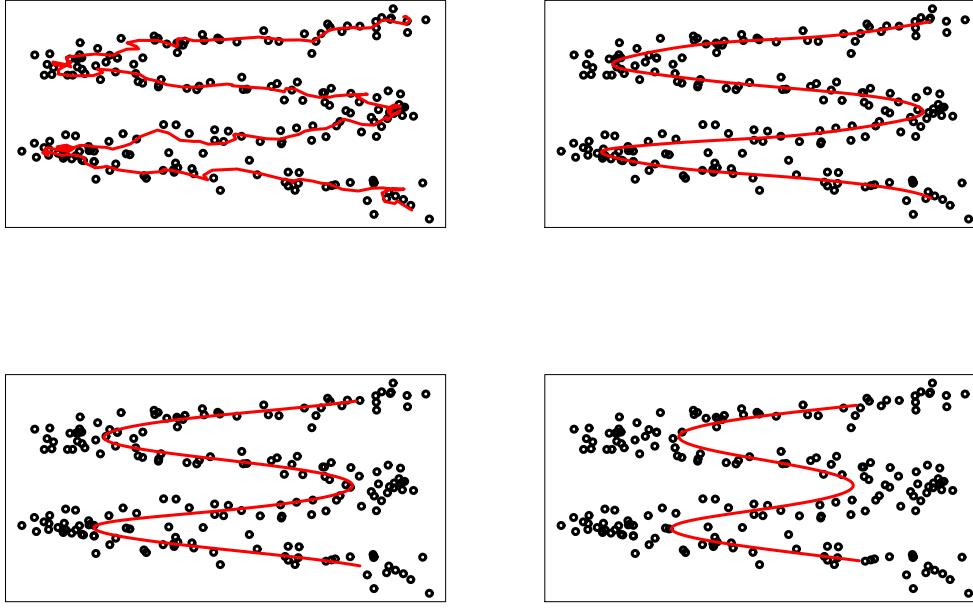


Figure 11: Principal curve with increasing amounts of regularization.

Theorem 7 Let f_* minimize $R(f)$ over \mathcal{F} . Assume that the distribution of X_i is supported on a compact set S and let $C = \sup_{x,x' \in S} \|x - x'\|^2$. For every $\epsilon > 0$

$$\mathbb{P} \left(|\widehat{R}(\widehat{f}) - R(f_*)| > 2\epsilon \right) \leq 2N \left(\frac{\epsilon}{4L}, \mathcal{F}, \|\cdot\|_\# \right) e^{-n\epsilon^2/(2C)}$$

for some constant L .

Proof. As with any of our previous risk minimization proofs, it suffices to show that

$$\mathbb{P} \left(\sup_{f \in \mathcal{F}} |\widehat{R}(f) - R(f)| > \epsilon \right) \leq 2N \left(\frac{\epsilon}{4L}, \mathcal{F}, \|\cdot\|_\# \right) e^{-n\epsilon^2/(2C)}.$$

Define $h_f(x) = \min_z \|x - f(z)\|$. For any fixed f , $\widehat{R}(f) - R(f) = P_n(h_f) - P(h_f)$ and so, by Hoeffding's inequality,

$$\mathbb{P} \left(|\widehat{R}(f) - R(f)| > \epsilon \right) \leq 2e^{-2n\epsilon^2/C}.$$

Let \mathcal{G} be the set of functions of the form $g_f(x, z) = \|x - f(z)\|^2$. Define a metric on \mathcal{G} by

$$d(g, g') = \sup_{z \in [0,1]^k, x \in S} |g_f(x, z) - g_{f'}(x, z)|.$$

Since S is compact, there exists $L > 0$ such that

$$\left| \|x - x'\|^2 - \|x - x''\|^2 \right| \leq L \|x' - x''\|$$

for all $x, x', x'' \in S$. It follows that

$$d(g, g') \leq L \sup_{z \in [0,1]^k} \|f(z) - f'(z)\| = L\|f - f'\|_\# . \quad (16)$$

Let $\delta = \epsilon/2$ and let f_1, \dots, f_N be an $\delta/2$ of \mathcal{F} . Let $g_j = g_{f_j}$, $j = 1, \dots, N$. It follows from (16) that g_1, \dots, g_N is an $\delta/(2L)$ cover of \mathcal{G} . For any f there exists f_j such that $d(g_f, g_j) \leq \delta/2$. So

$$\begin{aligned} |R(f) - R(f_j)| &= \left| \mathbb{E} \left(\inf_z \|X - f(z)\|^2 - \inf_z \|X - f_j(z)\|^2 \right) \right| \\ &= \left| \mathbb{E} \left(\inf_z g_f(X, z) - \inf_z g_j(X, z) \right) \right| \\ &\leq \mathbb{E} \left| \inf_z g_f(X, z) - \inf_z g_j(X, z) \right| \leq \delta/2. \end{aligned}$$

Similarly for \widehat{R} . So,

$$|\widehat{R}(f) - R(f)| \leq |\widehat{R}(f_j) - R(f_j)| + \delta.$$

Therefore,

$$\begin{aligned} \mathbb{P} \left(\sup_{f \in \mathcal{F}} |\widehat{R}(f) - R(f)| > \epsilon \right) &\leq \mathbb{P} \left(\max_{f_j} |\widehat{R}(f_j) - R(f_j)| > \epsilon/2 \right) \\ &\leq 2N \left(\frac{\epsilon}{4L}, \mathcal{F}, \|\cdot\|_\# \right) e^{-n\epsilon^2/(2C)}. \end{aligned}$$

□

Some comments on this result are in order. First, Smola, Mika, Schölkopf, Williamson (2001) compute $N(\frac{\epsilon}{4L}, \mathcal{F}, \|\cdot\|_\#)$ for several classes. For the Gaussian kernel they show that

$$N(\epsilon, \mathcal{F}, \|\cdot\|_\#) = O \left(\frac{1}{\epsilon} \right)^s$$

for some constant s . This implies that $R(\widehat{f}) - R(f_*) = O(n^{-1/2})$ which is a parametric rate of convergence. This is somewhat misleading. As we get more and more data, we should regularize less and less if we want a truly nonparametric analysis. This is ignored in the analysis above.

1.9 Random Projections: Part I

A simple method for reducing the dimension is to do a random projection. Surprisingly, this can actually preserve pairwise distances. This fact is known as the Johnson-Lindenstrauss Lemma, and this section is devoted to an elementary proof of this result.¹

¹In this section and the next, we follow some lecture notes by Martin Wainwright.

Let X_1, \dots, X_n be a dataset with $X_i \in \mathbb{R}^d$. Let S be a $m \times d$ matrix filled with iid $N(0, 1)$ entries, where $m < d$. Define

$$L(x) = \frac{Sx}{\sqrt{m}}.$$

The matrix S is called a *sketching matrix*. Define $Y_i = L(X_i)$ and note that $Y_i \in \mathbb{R}^m$. The projected dataset Y_1, \dots, Y_n is lower dimensional.

Theorem 8 (Johnson-Lindenstrauss) *Fix $\epsilon > 0$. Let $m \geq 32 \log n / \epsilon^2$. Then, with probability at least $1 - e^{-m\epsilon^2/16} \geq 1 - (1/n)^2$, we have*

$$(1 - \epsilon) \|X_i - X_j\|^2 \leq \|Y_i - Y_j\|^2 \leq (1 + \epsilon) \|X_i - X_j\|^2 \quad (17)$$

for all i, j .

Notice that the embedding dimension m , does not depend on the original dimension d .

Proof. For any $j \neq k$,

$$\frac{\|Y_j - Y_k\|^2}{\|X_i - X_j\|^2} - 1 = \frac{\|S(X_j - X_k)\|^2}{m\|X_i - X_j\|^2} - 1 = \frac{1}{m} \sum_{i=1}^m Z_i^2 - 1$$

where

$$Z_i = \left\langle S_i, \frac{X_j - X_k}{\|X_j - X_k\|} \right\rangle$$

where S_i is the i^{th} row of S . Note that $Z_i \sim N(0, 1)$ and so $Z_i^2 \sim \chi_1^2$ and $\mathbb{E}[Z_i^2] = 1$. The moment generating function of Z_i^2 is $m(\lambda) = (1 - 2\lambda)^{-1/2}$ (for $\lambda < 1/2$). So, for $\lambda > 0$ small enough,

$$\mathbb{E}[e^{\lambda(Z_i^2 - 1)}] = \frac{e^{-\lambda}}{\sqrt{1 - 2\lambda}} \leq e^{2\lambda^2}.$$

Hence,

$$\mathbb{E} \left[\exp \left(\lambda \sum_i (Z_i^2 - 1) \right) \right] \leq e^{2m\lambda^2}.$$

Thus

$$\begin{aligned} \mathbb{P} \left(\frac{1}{m} \sum_{i=1}^m Z_i^2 - 1 \geq \epsilon \right) &= \mathbb{P} \left(e^{\lambda \sum_{i=1}^m Z_i^2 - 1} \geq e^{\lambda m \epsilon} \right) \\ &\leq e^{-\lambda m \epsilon} \mathbb{E} \left(e^{\lambda \sum_{i=1}^m Z_i^2 - 1} \right) \leq e^{2m\lambda^2 - m\epsilon\lambda} \\ &\leq e^{-m\epsilon^2/8} \end{aligned}$$

where, in the last step, we chose $\lambda = \epsilon/4$. By a similar argument, we can bound $\mathbb{P} \left(\frac{1}{m} \sum_{i=1}^m Z_i^2 - 1 \leq -\epsilon \right)$. Hence,

$$\mathbb{P} \left(\left| \frac{\|S(X_j - X_k)\|^2}{m\|X_j - X_k\|^2} - 1 \right| \geq \epsilon \right) \leq 2e^{-m\epsilon^2/8}.$$

By the union bound, the probability that (17) fails for some pair is at most

$$n^2 2e^{-m\epsilon^2/8} \leq e^{-m\epsilon^2/16}$$

where we used the fact that $m \geq 32 \log n / \epsilon^2$. \square

1.10 Random Projections: Part II

The key to the Johnson-Lindenstrauss (JL) theorem was applying concentration of measure to the quantity

$$\Gamma(\mathcal{K}) = \sup_{u \in \mathcal{K}} \left| \frac{\|Su\|^2}{m} - 1 \right|$$

where

$$\mathcal{K} = \left\{ \frac{X_j - X_k}{\|X_j - X_k\|} : j \neq k \right\}.$$

Note that \mathcal{K} is a subset of the sphere \mathcal{S}^{d-1} .

We can generalize this to other subsets of the sphere. For example, suppose that we take $\mathcal{K} = \mathcal{S}^{d-1}$. Let $\widehat{\Sigma} = m^{-1}S^T S$. Note that each row of S_i has mean 0 and variance matrix I and $\widehat{\Sigma}$ is the estimate of the covariance matrix. Then

$$\begin{aligned} \sup_{u \in \mathcal{K}} \left| \frac{\|Su\|^2}{m} - 1 \right| &= \sup_{\|u\|=1} \left| \frac{\|Su\|^2}{m} - 1 \right| \\ &= \sup_{\|u\|=1} |u^T(m^{-1}S^T S - I)u| = \|\widehat{\Sigma} - I\| \end{aligned}$$

which is the operator norm of the difference between the sample covariance and true covariance.

Now consider least squares. Suppose we want to minimize $\|Y - X\beta\|^2$ where Y is an $n \times 1$ vector and X is a $n \times d$ matrix. If n is large, this may be expensive. We could try to approximate the solution by minimizing $\|S(Y - X\beta)\|^2$. The true least squares solution lies in the column space of X . The approximate solution will lie in the column space of SX . It can be shown that if This suggests taking

$$\mathcal{K} = \left\{ u \in \mathcal{S}^{d-1} : u = Xv \text{ for some } v \in \mathbb{R}^d \right\}.$$

Later we will show that, if $\Gamma(\mathcal{K})$ is small, then the solution to the reduced problem approximates the original problem.

How can we bound $\Gamma(\mathcal{K})$? To answer this, we use the *Gaussian width* which is defined by

$$W(\mathcal{K}) = \mathbb{E} \left[\sup_{u \in \mathcal{K}} \langle u, Z \rangle \right]$$

where $Z \sim N(0, I)$ and I is the $d \times d$ identity matrix.

Theorem 9 Let S be a $m \times d$ Gaussian projection matrix. Let \mathcal{K} be any subset of the sphere and suppose that $m \geq W^2(\mathcal{K})$. Then, for any $\epsilon \in (0, 1/2)$,

$$\mathbb{P} \left(\Gamma(\mathcal{K}) \geq 4 \left(\frac{W(\mathcal{K})}{\sqrt{m}} + \epsilon \right) \right) \leq 2e^{-m\epsilon^2/2}.$$

In particular, if $m \geq W^2(\mathcal{K})/\delta^2$, then $\Gamma(\mathcal{K}) \leq 8\delta$ with high probability.

Let us return to the JL theorem. In this case,

$$\mathcal{K} = \left\{ \frac{X_j - X_k}{\|X_j - X_k\|} : j \neq k \right\}.$$

In this case \mathcal{K} is finite. The number of elements is $N = \binom{n}{2}$. Note that $\log N \leq 2 \log n$. Since the set is finite, we know from our previous results on expectations of maxima, that

$$W(\mathcal{K}) \leq \sqrt{2 \log N} \leq \sqrt{4 \log n}.$$

According to the above theorem, we need to take $m \geq W^2/\delta^2 \geq \log n/\delta^2$ which agrees with the JL theorem.

The proof of the theorem is quite long but it basically uses concentration of measure arguments to control the maximum fluctuations as u varies over \mathcal{K} . When applied to least squares, if we want to approximate $\|Y - X\beta\|^2$ it turns out that the Gaussian width has constant order. Thus, taking m to be a large, fixed constant is enough. But this result assumed we are interested in approximating β , we need $m \approx n$ which is not useful. However, there is an improvement that uses iterative sketching that only requires $m = O(\log n)$ observations. A good reference is:

M. Pilanci and M. J. Wainwright. Iterative Hessian Sketch: Fast and accurate solution approximation for constrained least-squares. arXiv:1411.0347.

2 Estimating Low Dimensional Structure

Let $Y_1, \dots, Y_n \sim P$. We can think of the structure we are looking for as a function of P . Examples of such functions include:

- $T(P)$ = the support of P
- $T(P)$ = ridges of the density p
- $T(P)$ = dimension of the support
- $T(P)$ = DTM (distance to a measure)
- $T(P)$ = persistent homology of DTM.

A common example is when the support of P is a manifold M . In that case, we define the minimax risk

$$R_n = \inf_{\widehat{M}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[H(\widehat{M}, M(P))]$$

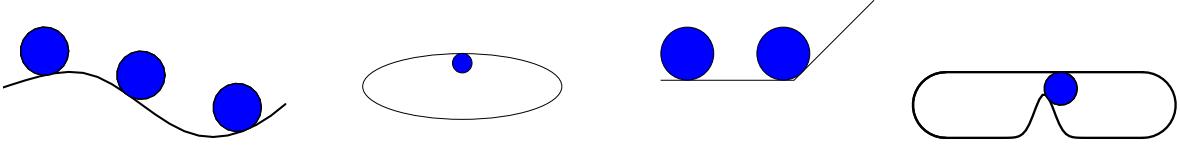


Figure 12: First two plots: a ball of radius $r < \kappa$ rolls freely. Third plot: ball cannot roll because reach is 0. Fourth: ball cannot roll because $r > \kappa$.

where H is the Hausdorff distance:

$$H(A, B) = \inf\{\epsilon : A \subset B \oplus \epsilon \text{ and } B \subset A \oplus \epsilon\}$$

and

$$A \oplus \epsilon = \bigcup_{x \in A} B(x, \epsilon).$$

2.1 Manifolds

A common starting place is to assume that P is supported on a manifold M . This is usually a bogus assumption. More realistically, the data might be concentrated near a low dimensional structure. Assuming that the structure is smooth and that the support is exactly on this structure is unrealistic. But it is a starting place. So, for now, assume that $Y_i \in \mathbb{R}^D$ and that P is supported on a manifold M of dimension $d < D$.

Just as we needed some conditions on a density function or regression function to estimate it, we needed a condition on a manifold to estimate it. The most common condition is that M has positive reach. The *reach* of a manifold M is the largest r such that $d(x, M) \leq r$ implies that x has a unique projection onto M . This is also called the thickness or condition number of the manifold; see Niyoki, Smale, and Weinberger (2009). Intuitively, a manifold M with $\text{reach}(M) = \kappa$ has two constraints:

1. Curvature. A ball of radius $r \leq \kappa$ can roll freely and smoothly over M , but a ball of radius $r > \kappa$ cannot.
2. Separation. M is at least 2κ from self-intersecting.

See Figure 12. Also, normal vectors of length less than κ will not cross. See Figure 13.

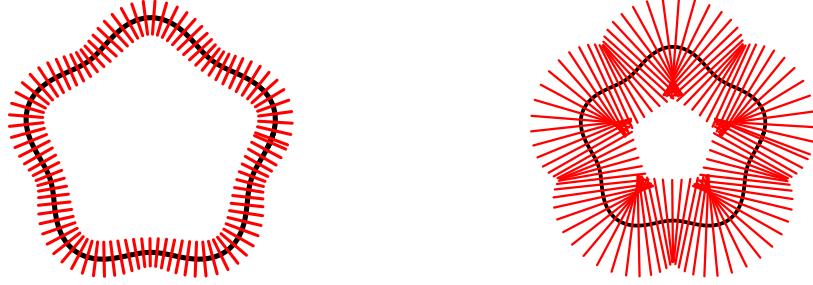


Figure 13: Left: Normal vectors of length $r < \kappa$ don't cross. Right: Normal vectors of length $r > \kappa$ do cross.

The easiest way to estimate a d -manifold embedded in \mathbb{R}^D is just to estimate the support of P . For example, the Devroye-Wise (1980) estimator is

$$\widehat{M} = \bigcup_i B(Y_i, \epsilon).$$

Choosing $\epsilon_n \asymp (1/n)^{1/D}$ we get

$$\mathbb{E}[H(\widehat{M}, M)] \leq \left(\frac{C \log n}{n} \right)^{\frac{1}{D}}.$$

This estimator is simple but sub-optimal. Note that the rate depends on the ambient dimension.

Let $Y_1, \dots, Y_n \sim P$ where

$$Y_i = \xi_i + Z_i$$

where $Y_i \in \mathbb{R}^D$, $\xi_1, \dots, \xi_n \sim G$ where G is uniform on a d -manifold M and the noise Z_i is perpendicular to M (uniform on the normals). It's a weird model but it was used in Niyogi, Smale, Weinberger (2008). Let \mathcal{P} be the set of distributions with bounded density on d -manifolds with reach at least κ . Then (GPVW 2011)

$$R_n = c \left(\frac{\log n}{n} \right)^{\frac{2}{2+d}}.$$

Thus the rate depends on d not D . I don't know a practical estimator to achieve this rate.

Now suppose that

$$Y_1, \dots, Y_n \sim (1 - \pi)U + \pi G$$

where G is supported on M , $0 < \pi \leq 1$, U is uniform on a compact set $\mathcal{K} \subset \mathbb{R}^D$. Then (GPVW 2012)

$$R_n \asymp \left(\frac{1}{n} \right)^{\frac{2}{d}}.$$

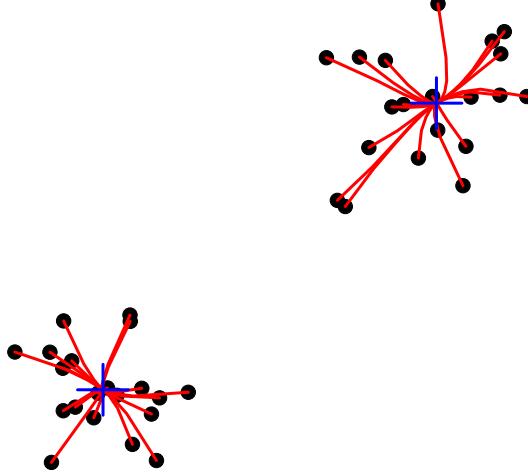


Figure 14: The mean shift algorithm.

A more realistic model is $Y_i = X_i + Z_i$ where $X_1, \dots, X_n \sim G$ and $Z_i \sim N(0, \sigma^2 I_D)$. Then

$$\frac{1}{\log n} \leq R_n \leq \frac{1}{\sqrt{\log n}}.$$

This means that, with additive noise, the problem is hopeless.

One solution is to give up on estimating M and instead estimate some approximation to M . This is the next topic.

2.2 Ridges

A ridge is a high-density, low dimensional structure. A 0-dimensional ridge is just a mode. In this case

$$\nabla p(x) = 0 \quad \text{and} \quad \lambda_{\max}(H(x)) < 0$$

where H is the Hessian. (assuming p is Morse). Recall that a mode can also be thought of as the destination of a gradient ascent path, π_x : i.e.

$$m = \lim_{t \rightarrow \infty} \pi_x(t)$$

where

$$\pi'_x(t) = \nabla p(\pi_x(t)).$$

The modes of p can be found by the mean-shift algorithm as in Figure 14.

Higher dimensional ridges can be defined as the zeros of a projected gradient. Think of the ridge of a mountain. The left plot in Figure 15 shows a density with a sharp, one-dimensional ridge. The right plot show the underlying manifold, the ridge, and the ridge of the smoothed density.

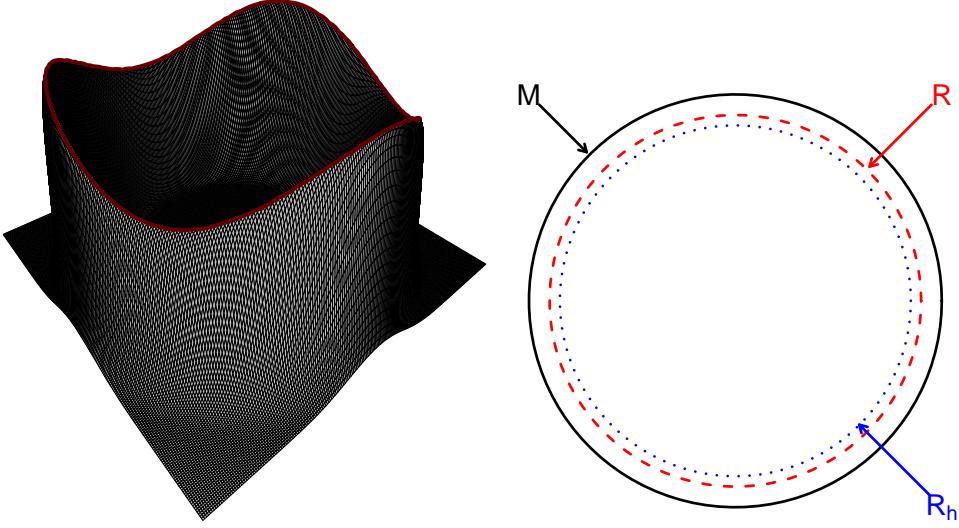


Figure 15: *Left:* The one-dimensional ridge of a density. *Right:* the manifold, the ridge of the density p , and the ridge of the smoothed density $p \star K_h$.

To define the ridge formally, let p be a density with gradient g and Hessian H . Denote the eigenvalues of $H(x)$ by

$$\lambda_1(x) \geq \lambda_2(x) \geq \cdots \geq \lambda_d(x) \geq \lambda_{d+1}(x) \geq \cdots \geq \lambda_D(x).$$

Let $U(x) = [W(x) : V(x)]$ be the matrix of eigenvectors. Then $L(x) = V(x)V^T(x)$ is the projector onto the local tangent space. Define the projected gradient $G(x) = L(x)g(x)$. Finally, define the ridge by

$$R(p) = \left\{ x : \lambda_{d+1}(x) < 0 \quad \text{and} \quad G(x) = 0 \right\}.$$

Several other definitions of a ridge have been proposed in the literature; see Eberly (1996). The one we use has several useful properties: if \hat{p} is close to p then $R(\hat{p})$ is close in Hausdorff distance to $R(p)$. If the data are sampled from a manifold M plus noise, then R is close to M and R is homotopic to M . That is: $Y_i = X_i + \epsilon_i$, where $X_1, \dots, X_n \sim G$, G is supported on M , $\epsilon_i \sim N(0, \sigma^2)$ and σ is small enough, then ridge R is homotopic to M . And, there is an algorithm to find the ridge: the subspace-constrained mean-shift algorithm (SCMS, Ozertem and Erdogmus 2011). (The usual mean-shift algorithm with a projection step.)

To estimate $R(p)$, estimate the density, its gradient, and its Hessian:

$$\hat{p}(y) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^D} K\left(\frac{y - Y_i}{h}\right)$$

\hat{g} = gradient of \hat{p} and \hat{H} = Hessian of \hat{p} . Denoising: remove low density points. Apply the SCMS algorithm.

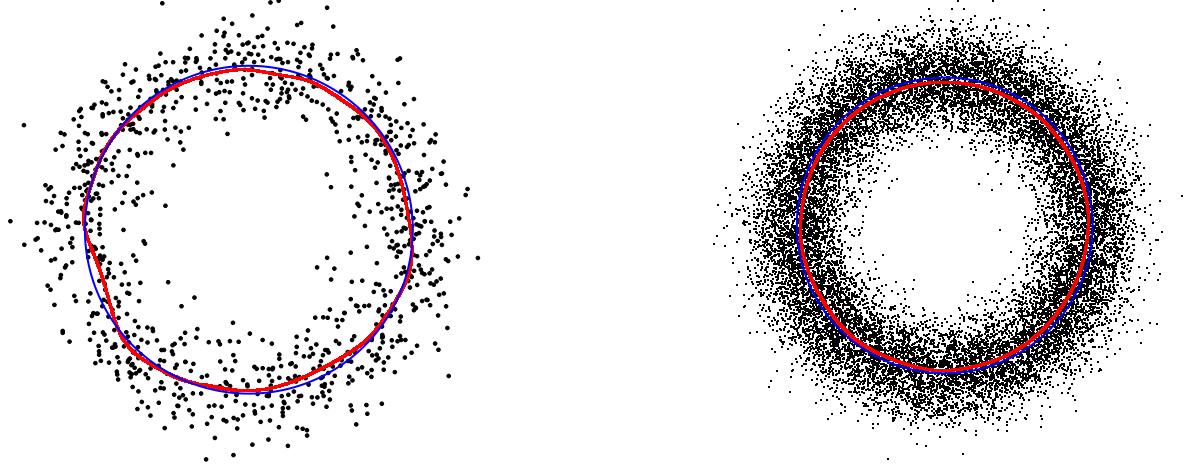


Figure 16: *Left:* Manifold in blue. Estimated ridge in red. *Right:* sample example with more data.

\widehat{R} is a consistent estimator of R and:

$$H(R, \widehat{R}) = O_P\left(n^{-\frac{2}{8+D}}\right)$$

For fixed bandwidth h (which still captures the shape),

$$H(R_h, \widehat{R}_h) = O_P\left(\sqrt{\frac{\log n}{n}}\right)$$

and \widehat{R}_h is (nearly) homotopic to R_h . See Figures 16 and 17 for examples. A real example is shown in 18 (from Chen, Ho, Freeman, Genovese and Wasserman: arXiv:1501.05303).

How to choose a good bandwidth h is not clear. Figure 19 shows that the ridge is fairly stable as we decrease h until we reach a phase transition where the ridge falls apart.

Large Sample Theory. Confidence sets for ridges can be computed using large sample theory (Chen, Genovese and Wasserman 2015). We have

$$\sup_t \left| \mathbb{P} \left\{ \sqrt{nh^{d+2}} H(\widehat{R}, R) \leq t \right\} - \mathbb{P} \left\{ \sup_{f \in \mathcal{F}} \|\mathbb{B}(f)\| \leq t \right\} \right| \leq \frac{C\sqrt{\log n}}{(nh^{d+2})^{1/8}},$$

where \mathcal{F} is a class of functions and \mathbb{B} is a Gaussian process on \mathcal{F} . Furthermore,

$$\sup_t |\widehat{F}_n(t) - F_n(t)| \leq \frac{C\sqrt{\log n}}{(nh^{d+2})^{1/8}}$$

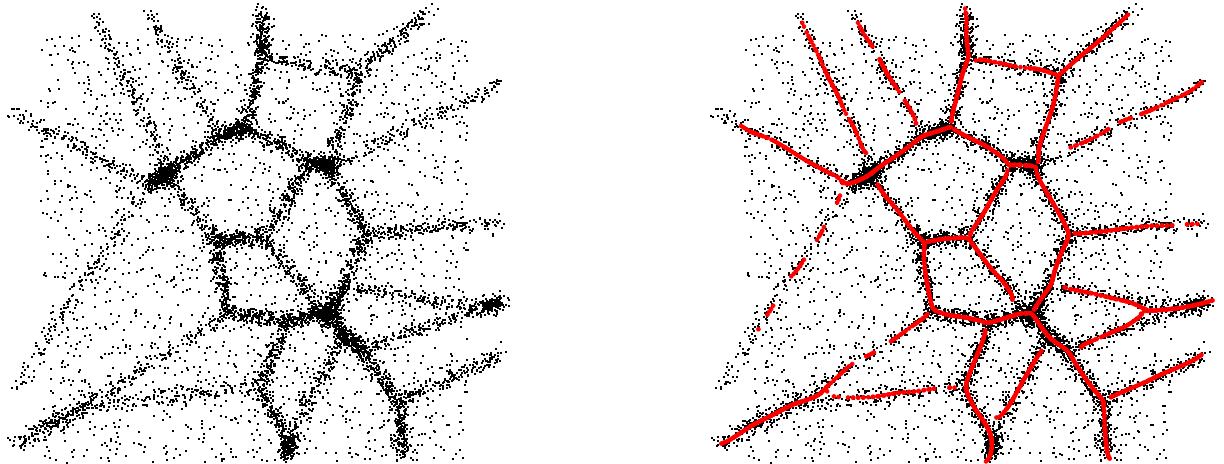


Figure 17: *Left: data. Right: SCMS output.*

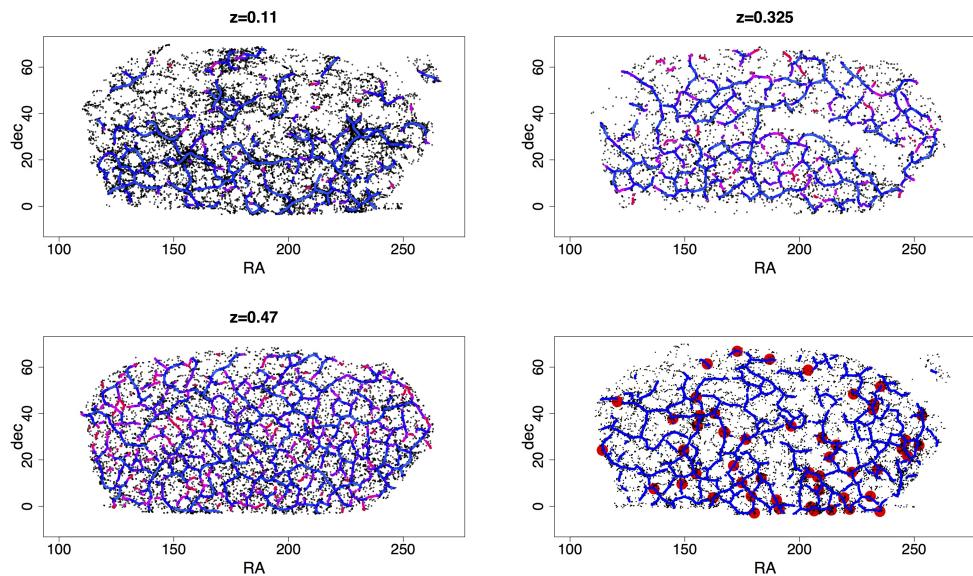


Figure 18: *Galaxy data from the Sloan Digital Sky Survey at three different redshifts. The fourth plot shows known galaxy clusters. From: Chen, Ho, Freeman, Genovese and Wasserman: arXiv:1501.05303*

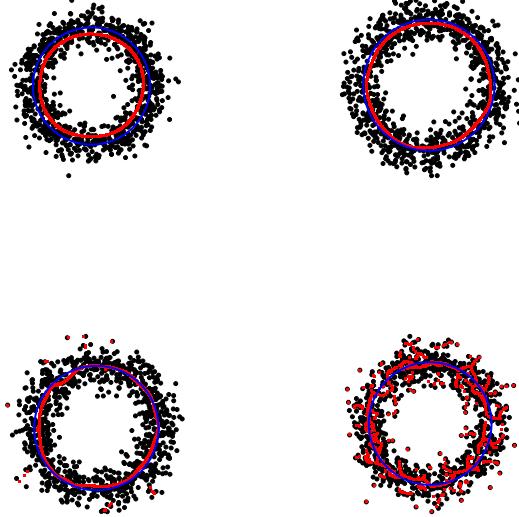


Figure 19: As we decrease the bandwidth, the ridge is quite stable. Eventually we reach a phase transition where the estimated ridge falls apart.

where

$$F_n(t) = \mathbb{P} \left\{ \sqrt{nh^{d+2}} H(\widehat{R}, R) \leq t \right\}$$

$$\widehat{F}_n(t) = \mathbb{P} \left\{ \sqrt{nh^{d+2}} H(\widehat{R}^*, \widehat{R}) \leq t | X_1, \dots, X_n \right\}$$

and the asterisks denote bootstrap versions. As a consequence,

$$\mathbb{P}(R \subset \widehat{R} \oplus c) = 1 - \alpha + o(1)$$

where $c = \widehat{F}_n^{-1}(\alpha)$. See Figure 20 for an example.

2.3 Persistent Homology

Warning: weird, strange stuff in this section!

Another approach to extracting structure from data is *persistent homology*, part of TDA (topological data analysis).

We look for topological features — such as connected components, one-dimensional voids, two dimensional voids — as a function of a scale parameter. We then keep track of the

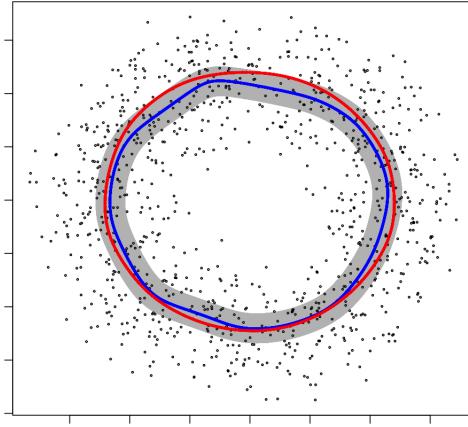


Figure 20: *Bootstrap confidence set for the ridge.*

birth and death of these features. The birth and death times are recorded on a *persistence diagram*.

Let S be a compact set. To describe the set, define the distance function $\Delta_S(x) = d(x, S) = \inf_{y \in S} \|x - y\|$. Let $L_t = \{x : \Delta_S(x) \leq t\}$ denote the lower level set. We call $\{L_t : t \geq 0\}$ a filtration. The persistence diagram D summarizes the topological features as a function of t . Figure 21 shows the distance function for a circle on the plane and Figure 22 shows some of the sub-level sets. Note there is one connected component and one void. But the void dies when t is large. Figure 23 shows the persistence diagram. The black dot shows that there is one connected component with birth time 0 and death time ∞ . The red triangle shows that there is a void with birth time 0 and death time 1.

Now suppose we have a sample $X_1, \dots, X_n \sim P$ where P is supported on some set S . Define

$$\Delta_n(x) = \min_i \|x - X_i\|.$$

The estimated sub-level sets are

$$\widehat{L}_t = \{x : \Delta_n(x) \leq t\} = \bigcup_{i=1}^n B(X_i, t).$$

So the union of balls = lower level sets of the empirical distance function.

Under very strong conditions,

$$\sup_x \|\Delta_n(x) - \Delta_S(x)\| \xrightarrow{P} 0$$

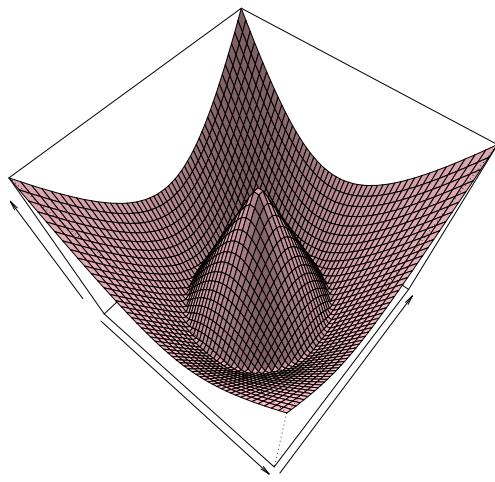


Figure 21: *Distance function for a circle in the plane.*

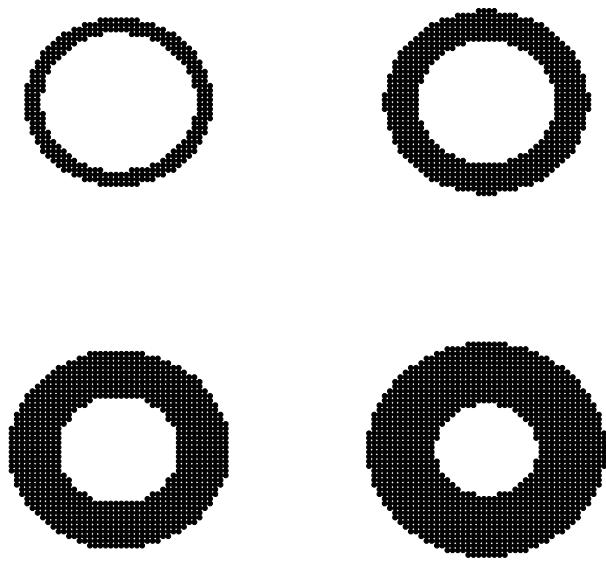


Figure 22: *Sub-level sets of the distance function for a circle in the plane.*

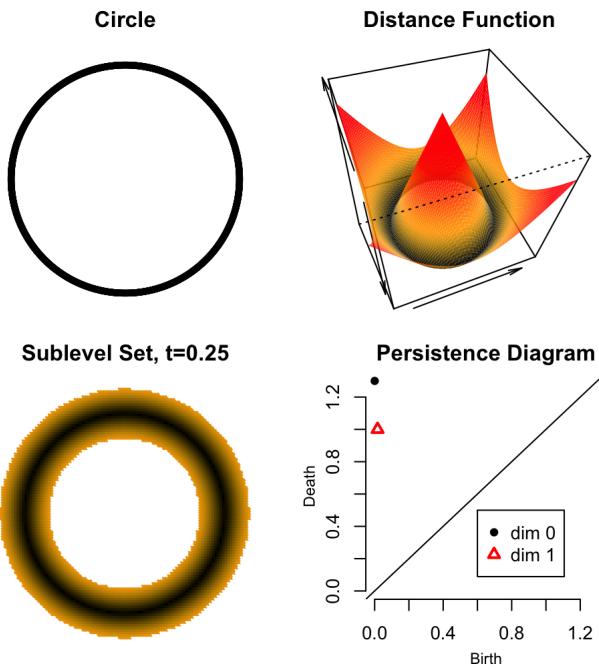


Figure 23: The circle S , the distance function, one typical sub-level set and the persistence diagram. The black dot shows that there is one connected component with birth time 0 and death time ∞ . The red triangle shows that there is a void with birth time 0 and death time 1.

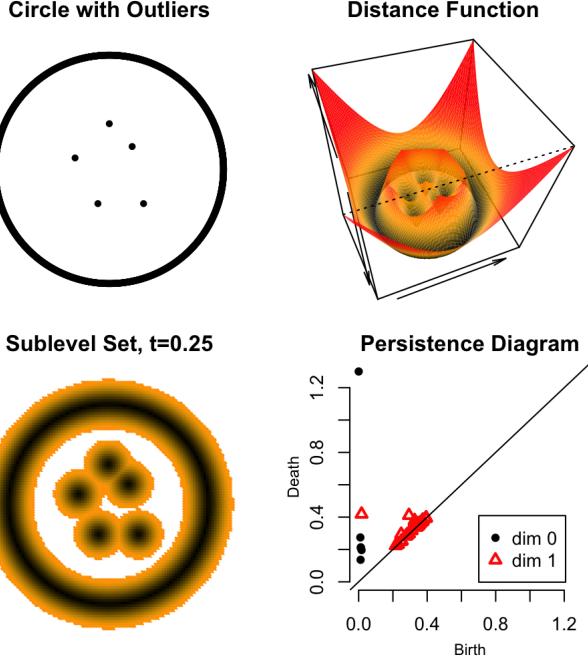


Figure 24: *Outliers are deadly if we use the empirical distance function.*

and this implies that

$$\text{bottleneck}(\widehat{D}, D) \xrightarrow{P} 0$$

where D is the persistence diagram, \widehat{D} is the estimated diagram and $\text{bottleneck}(\widehat{D}, D)$ is a metric between diagrams. (See Cohen-Steiner, Edelsbrunner and Harer 2007). But: if there is any noise or outliers, $\Delta_n(x)$ is a disaster! See Figure 24.

Can we make TDA more robust? There are two approaches. Replace the distance function with the DTM (distance to a measure) or use the upper level sets of a density estimate. The DTM was defined by Chazal, Cohen-Steiner and Merigot (2011). For each x , let $G_x(t) = P(\|X - x\| \leq t)$. Given $0 < m < 1$, the DTM is

$$\delta^2(x) = \frac{1}{m} \int_0^m [G_x^{-1}(u)]^2 du.$$

The sublevel sets of δ define a persistence diagram D . Let P_1 have DTM δ_1 with diagram D_1 and P_2 have DTM δ_2 with diagram D_2 . Then,

$$\text{bottleneck}(D_1, D_2) \leq \|\delta_1 - \delta_2\|_\infty.$$

The DTM has many nice properties: In particular, δ is distance like meaning that δ is 1-Lipschitz and δ^2 is 1-semiconcave.

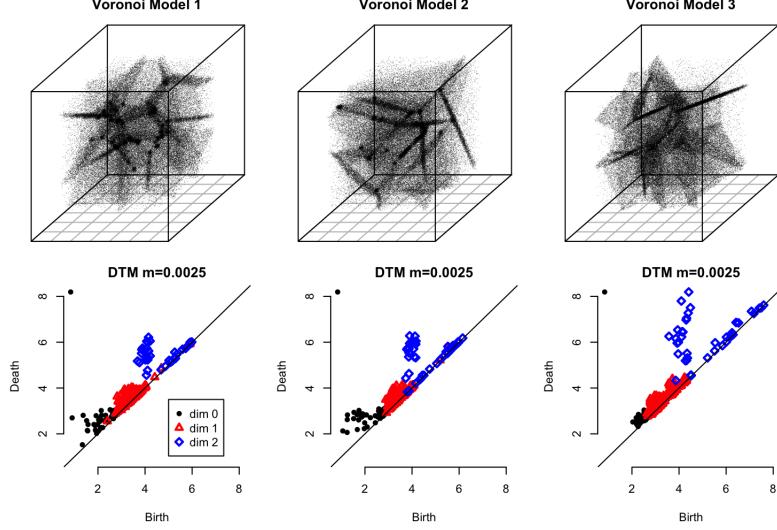


Figure 25: Examples of the DTM.

Note that the DTM $\delta(x) \equiv \delta_P(x)$ is a function of P . If we insert the empirical measure

$$P_n = \frac{1}{n} \sum_{i=1}^n \theta_{X_i}$$

where θ_x denotes a point mass at x , we get the plug-in estimator

$$\widehat{\delta}^2(x) = \left(\frac{1}{k_n} \right) \sum_{i=1}^{k_n} \|x - X_{(i)}\|^2$$

where $k_n = mn$ and $\|X_{(1)} - x\| \geq \|X_{(2)} - x\| \geq \dots$. Some examples are shown in Figures 25 and 26.

Under regularity conditions, we have that

$$\sqrt{n}(\widehat{\delta}^2(x) - \delta^2(x)) \rightsquigarrow \mathbb{B}(x)$$

where \mathbb{B} is a centered Gaussian process with covariance kernel

$$\kappa(x, y) = \frac{1}{m^2} \int_0^{F_x^{-1}(m)} \int_0^{F_y^{-1}(m)} \left(\mathbb{P}[B(x, \sqrt{t}) \cap B(y, \sqrt{s})] - F_x(t)F_y(s) \right) ds dt$$

and $F_x(t) = \mathbb{P}(\|X - x\|^2 \leq t)$. Recall the stability theorem:

$$\text{bottleneck}(\widehat{D}, D) \leq \sup_x \|\widehat{\delta}(x) - \delta(x)\|.$$

We can then use the bootstrap. Draw: $X_1^*, \dots, X_n^* \sim P_n$. Compute $\widehat{\delta}^*$ and repeat.

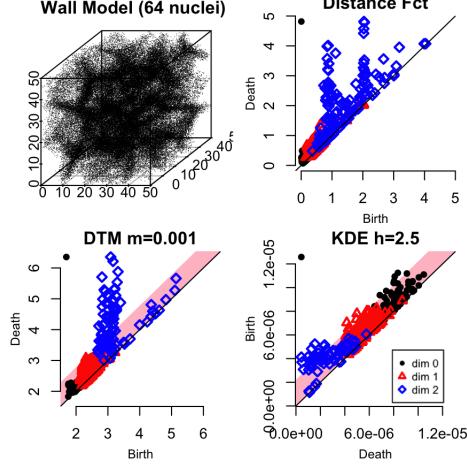


Figure 26: More examples of the DTM.

The map δ taking probability measures to DTM's is Hadamard differentiable. Hence, if we define \hat{c}_α by

$$\mathbb{P}(\sqrt{n}||\hat{\delta}^* - \hat{\delta}||_\infty > \hat{c}_\alpha | X_1, \dots, X_n) = \alpha.$$

Then

$$\mathbb{P}\left(||\delta - \hat{\delta}||_\infty \leq \frac{\hat{c}_\alpha}{\sqrt{n}}\right) \rightarrow 1 - \alpha.$$

A confidence set for true diagram D is

$$\mathcal{D} = \left\{ D : \text{bottleneck}(D, \hat{D}) \leq \frac{\hat{c}_\alpha}{\sqrt{n}} \right\}.$$

How to display this? Consider a feature (a point on the diagram) with birth and death time (b, d) . A feature is significant if it is not matched to the diagonal for any diagram in \mathcal{D} i.e. if

$$d - b > \frac{\hat{c}_\alpha}{\sqrt{n}}.$$

We can display this by adding a “noise band” on the diagram as in Figure 27.

As I mentioned before, we can also use the upper level sets of the kernel density estimator

$$\hat{p}_h(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^d} K\left(\frac{\|x - X_i\|}{h}\right)$$

which estimates $p_h(x) = \mathbb{E}[\hat{p}_h(x)]$. The upper-level sets $\{\hat{p}_h(x) > t\}$ define a persistence diagram \hat{D} . In TDA we do not have to let $h > 0$. This means that the rates are $O_P(1/\sqrt{n})$.

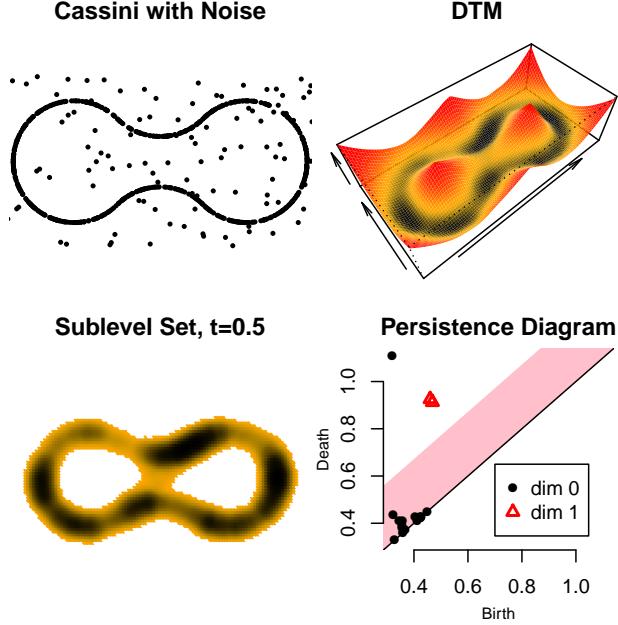


Figure 27: DTM, persistence diagram and significance band.

The diagram \widehat{D} of $\{\widehat{p}_h > t\}$ estimates the diagram D of $\{p_h > t\}$. Then

$$\text{bottleneck}(\widehat{D}, D) = O_P \left(\frac{1}{\sqrt{n}} \right).$$

We can view this from the RKHS point of view (Phillips, Wang and Zheng 2014). Define

$$\begin{aligned} D^2(P, Q) &= \int \int K_h(u, v) dP(u) dP(v) + \int \int K_h(u, v) dQ(u) dQ(v) \\ &\quad - 2 \int \int K_h(u, v) dP(u) dQ(v). \end{aligned}$$

Let θ_x be a point mass at x . Define

$$\begin{aligned} D^2(x) &\equiv D^2(P, \theta_x) \\ &= \int \int K_h(u, v) dP(u) dP(v) + K_h(x, x) - 2 \int K_h(x, u) dP(u) \end{aligned}$$

The plug-in estimator is

$$\widehat{D}^2(x) = \frac{1}{n^2} \sum_i \sum_j K_h(X_i, X_j) + K_h(x, x) - \frac{2}{n} \sum_i K_h(x, X_i).$$

The lower-level sets of \widehat{D} are (essentially) the same as the upper level sets of \widehat{p}_h . Now we proceed as with the DTM: get diagram, bootstrap etc. (Similar limiting theorems apply.)

The inferences are based on the stability theorem:

$$\text{bottleneck}(\widehat{D}, D) \leq \|\widehat{p}_h - p_h\|_\infty.$$

Now we can construct estimate, confidence band, etc. But sometimes $\text{bottleneck}(\widehat{D}, D) < \|\widehat{p}_h - p_h\|_\infty$. If we make slightly stronger assumptions, we get a better limiting result. Specifically,

$$\sqrt{n} \text{ bottleneck}(\widehat{D}, D) \rightsquigarrow \|Z\|_\infty$$

where, $Z \in \mathbb{R}^k$, $Z \sim N(0, \Sigma)$, and Σ is a function of the gradient and Hessian of p_h . This sidesteps the stability theorem. It is directly about the bottleneck distance.

We can then get a *bottleneck bootstrap*. Let

$$F_n(t) = \mathbb{P}(\sqrt{n} \text{ bottleneck}(\widehat{D}, D) \leq t).$$

Let $X_1^*, \dots, X_n^* \sim P_n$ where P_n is the empirical distribution. Let \widehat{D}^* be the diagram from \widehat{p}_h^* and let

$$\widehat{F}_n(t) = \mathbb{P}(\sqrt{n} \text{ bottleneck}(\widehat{D}^*, \widehat{D}) \mid X_1, \dots, X_n) \leq t)$$

be the bootstrap approximation to F_n . We have

$$\sup_t |\widehat{F}_n(t) - F_n(t)| \xrightarrow{P} 0.$$

So we can use $\widehat{c}_\alpha = \widehat{F}_n(1 - \alpha)/\sqrt{n}$. See Figure 28.

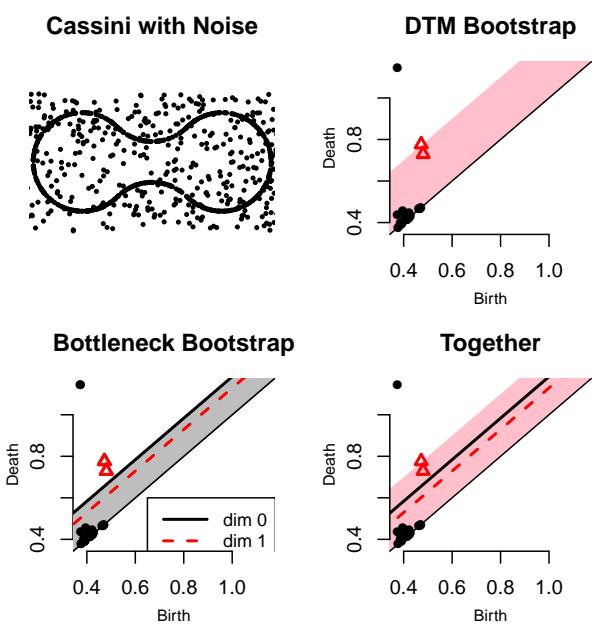


Figure 28: *The bottleneck bootstrap.*

Boosting

(Following Mohri, Rostamizadeh and Talwalkar.)

Let $Z_i = (X_i, Y_i)$ where $Y_i \in \{-1, +1\}$. Boosting is a way to combine *weak classifiers* into a better classifier. We make the weak learning assumption: for some $\gamma > 0$ we have an algorithm returns $h \in \mathcal{H}$ such that, for all P ,

$$P(R(h) \leq 1/2 - \gamma) \geq 1 - \delta$$

where $\gamma > 0$ is the edge.

Let us recall the AdaBboost algorithm:

1. Set $D_1(i) = 1/n$ for $i = 1, \dots, n$.
2. Repeat for $t = 1, \dots, T$:
 - (a) Let $h_t = \operatorname{argmin}_{h \in \mathcal{H}} P_{D_t}(Y_i \neq h(X_i))$.
 - (b) $\epsilon_t = P_{D_t}(Y_i \neq h_t(X_i))$.
 - (c) $\alpha_t = (1/2) \log((1 - \epsilon_t)/\epsilon_t)$.
 - (d) Let

$$D_{t+1}(i) = \frac{D_t(i) e^{-Y_i \alpha_t h_t(X_i)}}{Z_t}$$

where Z_t is a normalizing constant.

3. Set $g(x) = \sum_t \alpha_t h_t(x)$.
4. Return $h(x) = \operatorname{sign} g(x)$.

Training Error. Now we show that the training error decreases exponentially fast.

Lemma 1 *We have*

$$Z_t = 2\sqrt{\epsilon_t(1 - \epsilon_t)}.$$

Proof. Since $\sum_i D_t(i) = 1$ we have

$$\begin{aligned} Z_t &= \sum_i D_t(i) e^{-\alpha_t Y_i h_t(X_i)} = \sum_{Y_i h_t(X_i) = 1} D_t(i) e^{-\alpha_t} + \sum_{Y_i h_t(X_i) = -1} D_t(i) e^{\alpha_t} \\ &= (1 - \epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t} = 2\sqrt{\epsilon_t(1 - \epsilon_t)}. \end{aligned}$$

since $\alpha_t = (1/2) \log((1 - \epsilon_t)/\epsilon_t)$. \square

Theorem 2 Suppose that $\gamma \leq (1/2) - \epsilon_t$ for all t . Then

$$\widehat{R}(h) \leq e^{-2\gamma^2 T}.$$

Hence, the training error goes to 0 quickly.

Proof. Recall that $D_1(i) = 1/n$. So

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)e^{-\alpha_t Y_i h_t(X_i)}}{Z_t} = \frac{D_{t-1}(i)e^{-\alpha_{t-1} Y_i h_{t-1}(X_i)}e^{-\alpha_t Y_i h_t(X_i)}}{Z_t Z_{t-1}} \\ &= \dots = \frac{e^{-Y_i \sum_t \alpha_t h_t(X_i)}}{n \prod_t Z_t} = \frac{e^{-Y_i g(X_i)}}{n \prod_t Z_t} \end{aligned}$$

which implies that

$$e^{-Y_i g(X_i)} = n D_{T+1}(i) \prod_t Z_t. \quad (1)$$

Since $I(u \leq 0) \leq e^{-u}$ we have

$$\begin{aligned} \widehat{R}(h) &= \frac{1}{n} \sum_i I(Y_i g(X_i) \leq 0) \leq \frac{1}{n} \sum_i e^{-Y_i g(X_i)} = \frac{1}{n} \sum_i n (\prod_t Z_t) D_{T+1}(i) = \prod_{t=1}^T Z_t \\ &= \prod_t 2\sqrt{\epsilon_t(1-\epsilon_t)} = \prod_t \sqrt{1 - 4(1/2 - \epsilon_t)^2} \\ &\leq \prod_t e^{-2(1/2 - \epsilon_t)^2} \quad \text{since } 1-x \leq e^{-x} \\ &= e^{-2\sum_t (1/2 - \epsilon_t)^2} \leq e^{-2\gamma^2 T}. \end{aligned}$$

□

Generalization Error. The training error gets small very quickly. But how well do we do in terms of prediction error?

Let

$$\mathcal{F} = \left\{ \text{sign}\left(\sum_t \alpha_t h_t\right) : \alpha_t \in \mathbb{R}, h_t \in \mathcal{H} \right\}.$$

For fixed $h = (h_1, \dots, h_T)$ this is just a set of linear classifiers which has VC dimension T . So the shattering number is

$$\left(\frac{en}{T}\right)^T.$$

If \mathcal{H} is finite then the shattering number is

$$\left(\frac{en}{T}\right)^T \cdot |\mathcal{H}|^T.$$

If \mathcal{H} is infinite but has VC dimension d then the shattering number is bounded by

$$\left(\frac{en}{T}\right)^T \left(\frac{en}{d}\right)^{dT} \leq n^{Td}.$$

By the VC theorem, with probability at least $1 - \delta$,

$$R(\hat{h}) \leq \hat{R}(h) + \sqrt{\frac{T d \log n}{n}}.$$

Unfortunately this depends on T . We can fix this using margin theory.

Margins. Consider the classifier $h(x) = \text{sign}(g(x))$ where $g(x) = \sum_t \alpha_t h_t(x)$. The classifier is unchanged if we multiply g by a scalar. In particular, we can replace g with $\tilde{g} = g/\|\alpha\|_1$. This form of the classifier is a convex combination of the h_t 's.

We define the *margin at x* of $g = \sum_t \alpha_t h_t$ by

$$\rho(x) = \frac{yg(x)}{\|\alpha\|_1} = y\tilde{g}(x).$$

Think of $|\rho(x)|$ as our confidence in classifying x . The margin of g is defined to be

$$\rho = \min_i \rho(X_i) = \min_i \frac{Y_i g(X_i)}{\|\alpha\|_1}.$$

Note that $\rho \in [-1, 1]$.

To proceed we need to review Radamacher complexity. Given a class of functions \mathcal{F} with $-1 \leq f(x) \leq 1$ we define

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_i \sigma_i f(Z_i) \right]$$

where $P(\sigma_i = 1) = P(\sigma_i = -1) = 1/2$. If \mathcal{H} is finite then

$$\mathcal{R}_n(\mathcal{H}) \leq \sqrt{\frac{2 \log |\mathcal{H}|}{n}}.$$

If \mathcal{H} has VC dimension d then

$$\mathcal{R}_n(\mathcal{H}) \leq \sqrt{\frac{2d \log(en/d)}{n}}.$$

We will need the following two facts. First,

$$\mathcal{R}_n(\text{conv}(\mathcal{H})) = \mathcal{R}_n(\mathcal{H})$$

where $\text{conv}(\mathcal{H})$ is the convex hull of \mathcal{H} . Second, if

$$|\phi(x) - \phi(y)| \leq L||x - y||$$

for all x, y then

$$\mathcal{R}_n(\phi \circ \mathcal{F}) \leq L\mathcal{R}_n(\mathcal{F}).$$

The set of margin functions is

$$\mathcal{M} = \{yf(x) : f \in \text{conv}(\mathcal{H})\}.$$

We then have

$$\mathcal{R}_n(\mathcal{M}) = \mathcal{R}_n(\text{conv}(\mathcal{H})) = \mathcal{R}_n(\mathcal{H}).$$

A key result is that, with probability at least $1 - \delta$, for all $f \in \mathcal{F}$,

$$\mathbb{E}[f(Z)] \leq \frac{1}{n} \sum_i f(Z_i) + 2\mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{2 \log(1/\delta)}{n}}. \quad (2)$$

Now fix a number ρ and define the margin-sensitive loss function

$$\phi(u) = \begin{cases} 1 & u \leq 0 \\ 1 - \frac{u}{\rho} & 0 \leq u \leq \rho \\ 0 & u \geq \rho. \end{cases}$$

Note that

$$I(u \leq 0) \leq \phi(u) \leq I(u \leq \rho).$$

Assume that \mathcal{H} has VC dimension d . Then

$$\mathcal{R}_n(\phi \circ \mathcal{M}) \leq L\mathcal{R}_n(\mathcal{M}) \leq L\mathcal{R}_n(\mathcal{H}) \leq \frac{1}{\rho} \sqrt{\frac{2d \log(en/d)}{n}}.$$

Now define the empirical margin sensitive loss of a classifier f by

$$\hat{R}_\rho = \frac{1}{n} \sum_i I(Y_i f(X_i) \leq \rho).$$

Theorem 3 *With probability at least $1 - \delta$,*

$$R(g) \leq \hat{R}_\rho(g/||\alpha||_1) \leq \frac{1}{\rho} \sqrt{\frac{2d \log(en/d)}{n}} + \sqrt{\frac{2 \log(2/\delta)}{n}}.$$

Proof. Recall that $I(u \leq 0) \leq \phi(u) \leq I(u \leq \rho)$. Also recall that g and $\tilde{g} = g/\|\alpha\|_1$ are equivalent classifiers. Then using (2) we have

$$\begin{aligned} R(g) &= R(\tilde{g}) = P(Y\tilde{g}(X) \leq 0) \leq \frac{1}{n} \sum_i \phi(Y_i\tilde{g}(X_i)) + 2\mathcal{R}_n(\phi \circ \mathcal{M}) + \sqrt{\frac{2\log(2/\delta)}{n}} \\ &\leq \frac{1}{n} \sum_i \phi(Y_i\tilde{g}(X_i)) + \frac{1}{\rho} \sqrt{\frac{2d\log(en/d)}{n}} + \sqrt{\frac{2\log(2/\delta)}{n}} \\ &= \widehat{R}_\rho(g/\|\alpha\|_1) + \frac{1}{\rho} \sqrt{\frac{2d\log(en/d)}{n}} + \sqrt{\frac{2\log(2/\delta)}{n}}. \end{aligned}$$

□

Next we bound $\widehat{R}_\rho(g/\|\alpha\|_1)$.

Theorem 4 *We have*

$$\widehat{R}_\rho(g/\|\alpha\|_1) \leq \prod_{t=1}^T \sqrt{4\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}}.$$

Proof. Since $\phi(u) \leq I(u \leq \rho)$ we have

$$\begin{aligned} \widehat{R}_\rho(g/\|\alpha\|_1) &\leq \frac{1}{n} \sum_i I(Y_i g(X_i) - \rho\|\alpha\|_1 \leq 0) \\ &\leq e^{\rho\|\alpha\|_1} \frac{1}{n} \sum_i e^{-Y_i g(X_i)} \\ &= e^{\rho\|\alpha\|_1} \frac{1}{n} \sum_i n D_{T+1}(i) \prod_t Z_t = e^{\rho\|\alpha\|_1} \prod_t Z_t \\ &= \prod_{t=1}^T \sqrt{4\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}} \end{aligned}$$

since $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$ and $\alpha_t = (1/2)\log((1-\epsilon_t)/\epsilon_t)$. □

Assuming $\gamma \leq (1/2 - \epsilon_t)$ and $\rho < \gamma$ then it can be shown that $\sqrt{4\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}} \equiv b < 1$. So $\widehat{R}_\rho(g/\|\alpha\|_1) \leq b^T$. Combining with the previous result we have, with probability at least $1 - \delta$,

$$R(g) \leq b^T + \frac{1}{\rho} \sqrt{\frac{2d\log(en/d)}{n}} + \sqrt{\frac{2\log(2/\delta)}{n}}.$$

This shows that we get small error even with T large (unlike the earlier bound based only on VC theory).

Support Vector Machines

These notes are based on Mohri, Rostamizadeh and Talwalkar (2012).

Some Convex Optimization. Consider

$$\min_x f(x) \quad \text{subject to} \quad g_i(x) \leq 0 \quad i = 1, \dots, m.$$

Define the Lagrangian

$$\mathcal{L} = f(x) + \sum_j \alpha_j g_j(x).$$

The *dual function* is defined by

$$F(\alpha) = \inf_x \mathcal{L}.$$

A central result in convex optimization is that the original problem can be solved by maximizing F subject to $\alpha_i \geq 0$ and $\alpha_i g_i(x_i) = 0$.

Hyperplanes and SVM's. Suppose we have data $(X_1, Y_1), \dots, (X_n, Y_n)$ that can be separated by a hyperplane. Let $b + w^T x = 0$ be such a hyperplane. Note that $Y_i(b + X_i^T w) \geq 1$ for all i . Any re-scaled version of the hyper-plane is the same classifier. So re-scale the hyper-plane so that

$$\min_i |b + w^T X_i| = 1.$$

If x_0 is any point, then using some simple algebra, we find that the distance to the hyperplane is

$$\frac{|b + w^T x_0|}{\|w\|}.$$

We call the distance to the closest point, the *margin* ρ . Since $|\min_i |b + w^T X_i|| = 1$, we see that

$$\rho = \min_i \frac{|w^T X_i + b|}{\|w\|} = \frac{1}{\|w\|}.$$

The support vector machine (SVM) is the hyperplane that maximized the margin. But maximizing $1/\|w\|$ is the same as minimizing $\|w\|$ which is the same as minimizing $(1/2)\|w\|^2$. So finding the SVM corresponds to:

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 \quad \text{subject to } Y_i(w^T X_i + b) \geq 1 \quad i = 1, \dots, n.$$

The Lagrangian for this problem is

$$\mathcal{L} = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i [Y_i(w^T X_i + b) - 1]$$

where $\alpha_i \geq 0$ and $\alpha_i[Y_i(w^T X_i + b) - 1] = 0$. If we set $\nabla_w \mathcal{L} = 0$ and $\nabla_b \mathcal{L} = 0$ we get the two equations

$$w = \sum_i \alpha_i Y_i X_i = 0$$

$$0 = \sum_i \alpha_i Y_i.$$

If we insert $w = \sum_i \alpha_i Y_i X_i$ into \mathcal{L} and use the fact that $\sum_i \alpha_i Y_i = 0$ we get

$$\mathcal{L} = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j (X_i^T X_j).$$

This leads to the optimization

$$\text{maximize } \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j (X_i^T X_j)$$

subject to $\alpha_i \geq 0$ and $\alpha_i[Y_i(w^T X_i + b) - 1] = 0$. Note two importnat facts: (i) this is a quadratic program so it can be solved quickly and (ii) we don't need the X_i 's we only need the inner products $X_i^T X_j$.

Consider the constraint $\alpha_i[Y_i(w^T X_i + b) - 1] = 0$. If $\alpha_i > 0$ then $Y_i(w^T X_i + b) = 1$ which implies that this point lies on the boundary of the margin. Such a point is called a support vector. On the other hand, if $Y_i(w^T X_i + b) > 1$ then $\alpha_i = 0$. Since $w = \sum_i \alpha_i Y_i X_i$ this means that the hyperplane only depends on the support vectors.

If (X_i, Y_i) is a support vector then $W^T X_i + b = Y_i$. Since $w = \sum_j \alpha_j Y_j X_j$, we see that

$$b = Y_i - \sum_j \alpha_j Y_j X_j^T X_i.$$

Multiply by $\alpha_i Y_i$ and sum to get

$$\sum_i \alpha_i Y_i b = \sum_i \alpha_i Y_i^2 - \sum_{i,j} \alpha_i \alpha_j Y_i Y_j (X_i^T X_j).$$

Since $Y_i^2 = 1$, $w = \sum_i \alpha_i Y_i X_i$ and $\sum_i \alpha_i Y_i = 0$ this implies that

$$0 = \sum_i \alpha_i - \|w\|^2.$$

The margin ρ is $1/\|w\|$ so that

$$\rho^2 = \frac{1}{\|w\|^2} = \frac{1}{\sum_i \alpha_i} = \frac{1}{\|\alpha\|_1}.$$

The Non-separable Case. Usually, the data are not linearly separable. So we can't assume that $Y_i(w^T X_i + b) \geq 1$. We introduce slack variables $\xi_i \geq 0$ and instead require

$$Y_i(w^T X_i + b) \geq 1 - \xi_i.$$

This allows points to be incorrectly classified. But it also allows points to be correctly classified but be inside the margin. We change the optimization problem to

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

subject to $Y_i(w^T X_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$. The constant $C \geq 0$ controls the amount of slack that is allowed.

The Lagrangian is

$$\mathcal{L} = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha_i [Y_i(w^T X_i + b) - 1 + \xi_i] - \sum_i \beta_i \xi_i.$$

Setting the derivative to 0 leads to the conditions

$$\begin{aligned} w &= \sum_i \alpha_i Y_i X_i \\ 0 &= \sum_i \alpha_i Y_i \\ C &= \alpha_i + \beta_i \\ 0 &= \alpha_i \text{ or } Y_i(w^T X_i + b) = 1 - \xi_i \\ 0 &= \beta_i \text{ or } \xi_i = 0. \end{aligned}$$

When $\alpha_i > 0$ we call X_i a support vector. If $\alpha_i \neq 0$ then

$$Y_i(w^T X_i + b) = 1 - \xi_i.$$

If $\xi_i = 0$ then X_i lies on the marginal hyperplane. If $\xi_i \neq 0$ then $\beta_i = 0$ which implies $\alpha_i = C$. In summary, support vectors lie on the marginal hyperplane or $\alpha_i = C$.

The dual problem has a simple form:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j X_i^T X_j$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i Y_i = 0$. Again, it is a quadratic program and only involves inner products of the X_i .

Since the VC dimension of hyperplane classifiers is $d + 1$, we know that, with probability at least $1 - \delta$,

$$R(h) \leq R(\hat{h}) + \sqrt{\frac{2(d+1) \log(en/(n+1))}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}. \quad (1)$$

But this bound does not use the structure of SVM's. For this, we turn to margin theory.

Margins. Recall that the margin is

$$\rho = \min_i \frac{Y_i(w^T X_i + b)}{\|w\|}.$$

We can improve the VC bound using the margin.

Theorem 1 Suppose that the sample space is contained in $\{x : \|x\| \leq r\}$. Let \mathcal{H} be the set of hyperplanes satisfying $\|w\| \leq \Lambda$ and $\min_i |w^T X_i| = 1$. Then $\text{VC}(\mathcal{H}) \leq r^2 \Lambda^2$.

Proof. Suppose that $\{x_1, \dots, x_d\}$ can be shattered. Then for $y \in \{-1, +1\}^d$ there exists w such that $1 \leq y_i(w^T x_i)$ for all i . Sum over i to get

$$d \leq w^T \sum_i y_i x_i \leq \|w\| \left\| \sum_i y_i x_i \right\| \leq \Lambda \left\| \sum_i y_i x_i \right\|.$$

This holds for all choices of y_i . So it holds if Y_i is drawn uniformly over $\{-1, +1\}$. Thus $\mathbb{E}[Y_i Y_j] = \mathbb{E}[Y_i] \mathbb{E}[Y_j] = 0$ for $i \neq j$ and $\mathbb{E}[Y_i Y_i] = 1$. So

$$\begin{aligned} d &\leq \Lambda \mathbb{E} \left\| \sum_{i=1}^d Y_i x_i \right\| \leq \Lambda \sqrt{\mathbb{E} \left\| \sum_i Y_i x_i \right\|^2} \\ &= \Lambda \sqrt{\sum_{i,j} \mathbb{E}[Y_i Y_j] x_i^T x_j} = \Lambda \sqrt{\sum_i x_i^T x_i} \\ &\leq \Lambda \sqrt{dr^2} = \Lambda r \sqrt{d} \end{aligned}$$

so that $d \leq r^2 \Lambda$. \square

If the data are separable, the hyperplane satisfies $\|w\| = 1/\rho$ so that $\Lambda^2 = 1/\rho^2$ and hence $d \leq r^2/\rho^2$. Plugging this into (1) we get

$$R(h) \leq R(\hat{h}) + \sqrt{\frac{2r^2 \log((en\rho^2)/r^2)}{n\rho^2}} + \sqrt{\frac{\log(1/\delta)}{2n}} \quad (2)$$

which is dimension independent.

Nonparametric SVM's. We can get a nonparametric SVM using RKHS's by replacing x with a feature map $\Phi(x)$. Recall that $\Phi(x_1)^T \Phi(x_2) = K(x_1, x_2)$. So we get a nonparametric

SVM by solving

$$\max_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j K(X_i, X_j)$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i Y_i = 0$. The classifier is

$$h(x) = \text{sign} \left(\sum_i Y_i K(X_i, x) + b \right).$$

This is a nonlinear (nonparametric) classifier.

Online Learning

Once again, we follow Mohri, Rostamizadeh and Talwalkar (2012).

Online (sequential) prediction is amazing because it is completely assumption free. The basic setup is as follows:

1. For $t = 1, \dots, T$:
 - (a) Observe x_t .
 - (b) Predict \hat{y}_t .
 - (c) Observe y_t .
 - (d) Incur loss $L(\hat{y}_t, y_t)$.
2. The cumulative loss is $\sum_t L(\hat{y}_t, y_t)$.

Usually we will assume that $y_t \in \{0, 1\}$ and that $L(\hat{y}_t, y_t) = I(\hat{y}_t \neq y_t)$.

In the *expert advice* setting we have N algorithms (experts). The prediction from algorithm i is $y_{t,i}$. The goal in this case is to minimize the *regret*

$$R = \sum_t L(\hat{y}_t, y_t) - \min_i L(y_{t,i}, y_t).$$

Halving Algorithm. This is the simplest case. We have a finite set of predictors \mathcal{H} . We assume there is one $h \in \mathcal{H}$ that makes perfect predictions. Let $M(h)$ be the maximum number of mistakes that our algorithm makes (over all x_1, \dots, x_T). Let $M(\mathcal{H}) = \max_h M(h)$. The algorithm is as follows:

1. Set $\mathcal{H}_1 = \mathcal{H}$.
 - (a) Observe x_t . Let \hat{y}_t be the majority vote of \mathcal{H}_t .
 - (b) Observe y_t .
 - (c) If $\hat{y}_t \neq y_t$ set $\mathcal{H}_t = \{h : h(x_t) = y_t\}$.

Theorem 1 $M(\mathcal{H}) \leq \log_2 |\mathcal{H}|$.

Proof. If $\hat{y}_t \neq y_t$ then we reduce \mathcal{H}_t by at least half so that $|\mathcal{H}_{t+1}| \leq (1/2)|\mathcal{H}_t|$. So after $\log_2 |\mathcal{H}|$ mistakes there is only one expert left which must be the perfect expert and hence there will be no more mistakes. \square

The assumption of a perfect predictor is unrealistic so let's move on to a more realistic setting.

Weighted Majority. The algorithm is:

1. Set $\beta \in [0, 1)$.
2. Set $w_{1,i} = 1$ for $i = 1, \dots, N$.
3. For $t = 1, \dots, T$:

- (a) Observe x_t .
- (b) If

$$\sum_{y_{t,i}=1} w_{t,i} \geq \sum_{y_{t,i}=0} w_{t,i}$$

then $\hat{y}_t = 1$ else $\hat{y}_t = 0$.

- (c) Observe y_t .
- (d) If $\hat{y}_t \neq y_t$:

$$\begin{aligned} &\text{If } y_{t,i} \neq y_t \text{ set } w_{t+1,i} = \beta w_{t,i} \\ &\text{If } y_{t,i} = y_t \text{ set } w_{t+1,i} = w_{t,i}. \end{aligned}$$

Let $m^* = \min_i \sum_t I(y_{t,i} \neq y_t)$ be the loss of the best expert. Let m be the loss of the algorithm.

Theorem 2 We have that

$$m \leq \frac{\log N + m^* \log(1/\beta)}{\log(2/(1+\beta))}.$$

Proof. Let $W_t = \sum_i w_{t,i}$. Note that $W_1 = N$. Because of the weighted majority rule, we have that if there is an error,

$$W_{t+1} \leq \left(\frac{1}{2} + \frac{\beta}{2}\right) W_t = \left(\frac{1+\beta}{2}\right) W_t.$$

Hence,

$$W_T \leq \left(\frac{1+\beta}{2}\right)^m N.$$

On the other hand, for each i ,

$$W_T \geq w_{T,i} = \beta^{m(T,i)}$$

where $m(T, i)$ is the number of mistakes from expert i . In particular, this holds for the best expert so that

$$W_T \geq \beta^{m^*}.$$

Combining these two bounds,

$$\beta^{m^*} \leq W_T \leq \left(\frac{1+\beta}{2}\right)^m N.$$

Taking the log and re-arranging terms gives the result. \square

This is a nice result but it does not guarantee that the loss is small. To see this, suppose there are two experts. The first outputs $0, 0, \dots, 0$ and the second outputs $1, 1, \dots, 1$. Note that $m^* \leq T/2$. Now suppose that nature is evil and sets $y_t = 0$ when $\hat{y}_t = 1$ and sets $y_t = 1$ when $\hat{y}_t = 0$. Then $m = T$. So the regret is $R = m - m^* \geq T/2$. Can we make the regret smaller. Yes, as we now show.

Randomized Weighted Majority. For this algorithm we choose expert i with some probability $p_{t,i}$. We receive a vector of losses $\ell_t = (\ell_{t,1}, \dots, \ell_{t,N})$. The expected loss is $L_t = \sum_i p_{t,i} \ell_{t,i}$ and the cumulative expected loss is $\mathcal{L}_T = \sum_{t=1}^T L_t$. We also define $\mathcal{L}_{T,i} = \sum_t \ell_{t,i}$ and the minimum loss $\mathcal{L}^* = \min_i \mathcal{L}_{T,i}$. Here is the algorithm:

1. Set $w_{i,1} = 1$ for $i = 1, \dots, N$.
2. Set $p_{1,i} = 1/N$ for $i = 1, \dots, N$.
3. For $t = 1, \dots, T$:
 - (a) If $\ell_{t,i} = 1$ set $w_{t+1,i} = \beta w_{t,i}$. If $\ell_{t,i} = 0$ set $w_{t+1,i} = w_{t,i}$.
 - (b) Let $W_{t+1} = \sum_i w_{t+1,i}$.
 - (c) Set $p_{t+1,i} = w_{t+1,i}/W_{t+1}$.

Theorem 3 *We have*

$$\mathcal{L}_T \leq \mathcal{L}^* + 2\sqrt{T \log N}.$$

The remarkable thing about this result is that the regret only grows at rate \sqrt{T} . In other words, the average regrest is $\sqrt{\log N/T}$.

Proof. Set $p_{t,i} = w_{t,i}/W_t$ we have that $w_{t,i} = W_t p_{t,i}$. Hence,

$$\begin{aligned} W_{t+1} &= \sum_{i: \ell_{t,i}=0} w_{t,i} + \beta \sum_{i: \ell_{t,i}=1} w_{t,i} = W_t + (\beta - 1) \sum_{i: \ell_{t,i}=1} w_{t,i} \\ &= W_t + (\beta - 1) W_t \sum_{i: \ell_{t,i}=1} p_{t,i} = W_t + (\beta - 1) W_t L_t = W_t [1 - (1 - \beta) L_t]. \end{aligned}$$

Recalling that $W_1 = N$ we see that

$$W_{T+1} = N \prod_t [1 - (1 - \beta)L_t].$$

On the other hand,

$$W_{T+1} \geq \max_i w_{T+1,i} = \beta^{\mathcal{L}_*}.$$

Combining these inequalities we get

$$\beta^{\mathcal{L}_*} \leq W_{T+1} \leq N \prod_t [1 - (1 - \beta)L_t].$$

Hence,

$$\begin{aligned} \mathcal{L}_* \log \beta &\leq \log N + \sum_t [1 - (1 - \beta)L_t] \\ &\leq \log N - (1 - \beta) \sum_t L_t \quad \text{since } \log(1 - x) \leq -x \\ &= \log N - (1 - \beta)\mathcal{L}_T. \end{aligned}$$

Re-arranging terms we get

$$\mathcal{L}_T \leq \frac{\log N}{1 - \beta} + (1 - \beta)T + \mathcal{L}_*.$$

Now we set $\beta = 1 - \sqrt{\log N/T}$ and we have

$$\mathcal{L}_T \leq \mathcal{L}_* + 2\sqrt{T \log N}.$$

□

Exponential Weights. Now we allow the loss to take values in $[0, 1]$. We handle this case by modifying the weights. We assume that the loss function L is convex in its first argument. In what follows, $L_{t,i}$ is the total loss of expert i after t steps. Here is the algorithm:

1. Set $w_{1,i} = 1$ for $i = 1, \dots, N$.
2. For $t = 1, \dots, T$:

- (a) Observe x_t .
- (b) Let

$$\hat{y}_t = \frac{\sum_i w_{t,i} y_{t,i}}{\sum_i w_{t,i}}.$$

- (c) Observe y_t . Set

$$w_{t+1,i} = w_{t,i} e^{-\theta L(y_{t,i}, y_t)}.$$

Theorem 4 If $\theta = \sqrt{8 \log N/T}$ then the regret satisfies

$$R_T \leq \sqrt{T \log N/2}.$$

Remark: The interesting thing about the proof below is that it uses probabilistic ideas even though there is no probability distribution in the setup of the problem.

Proof. Let us begin by recalling the following fact: suppose that $a \leq X \leq b$ and $\mathbb{E}[X] = 0$. Then

$$\mathbb{E}[e^{tX}] \leq e^{t^2(b-a)^2/8}. \quad (1)$$

Define

$$\Phi_t = \log \sum_i w_{t,i}.$$

Then

$$\begin{aligned} \Phi_{t+1} - \Phi_t &= \log \frac{\sum_i w_{t+1,i}}{\sum_i w_{t,i}} = \log \frac{w_{t,i} e^{-\theta L(y_{t,i}, y_t)}}{\sum_i w_{t,i}} \\ &= \log \mathbb{E}_t e^{\theta X} \end{aligned}$$

where

$$X = -L(y_{t,i}, y_t) \in [-1, 0]$$

and \mathbb{E}_t refers to expectation with respect to the distribution with probability function $p_{t,i} = \frac{w_{t,i}}{\sum_i w_{t,i}}$. So

$$\begin{aligned} \Phi_{t+1} - \Phi_t &= \log \mathbb{E}_t e^{\theta(X - \mathbb{E}_t[X]) + \theta \mathbb{E}_t[X]} \\ &= \theta \mathbb{E}_t[X] + \log \mathbb{E}_t e^{\theta(X - \mathbb{E}_t[X])} \\ &\leq \theta \mathbb{E}_t[X] + \frac{\theta^2}{8} \quad \text{using (1)} \\ &= -\theta \mathbb{E}_t[L(y_{t,i}, y_t)] + \frac{\theta^2}{8} \\ &\leq -\theta L(\mathbb{E}_t[y_{t,i}], y_t) + \frac{\theta^2}{8} \quad \text{using convexity} \\ &= -\theta L(\hat{y}_t, y_t) + \frac{\theta^2}{8} \quad \text{definition of } \hat{y}_t. \end{aligned}$$

Now we sum over t to get

$$\Phi_{T+1} - \Phi_1 \leq -\theta \sum_t L(\hat{y}_t, y_t) + \frac{\theta^2 T}{8}.$$

Next we have the lower bound

$$\begin{aligned} \Phi_{T+1} - \Phi_1 &= \log \sum_i w_{T+1,i} - \log N = \log \sum_i e^{-\theta L_{T,i}} - \log N \\ &\geq \log \max_i e^{-\theta L_{T,i}} - \log N = -\theta \min_i L_{T,i} - \log N. \end{aligned}$$

Combining the lower and upper bound we have

$$-\theta \min_i L_{T,i} - \log N \leq \frac{\theta^2 T}{8} - \theta \sum_t L(\hat{y}_t, y_t)$$

which implies that

$$\sum_t L(\hat{y}_t, y_t) - \min_i L_{T,i} \leq \frac{\log N}{4} + \frac{\theta T}{8}.$$

The result follows by setting $\theta = \sqrt{8 \log N / T}$. \square

Online to Batch. The setting we have focused on in class is the batch setting where we observe random variables $(X_1, Y_1), \dots, (X_n, Y_n)$ from some distribution. It turns out that we can apply online algorithms to the batch setting. We again assume that the loss L is convex in its first argument.

Let \mathcal{H} be a set of classifiers and assume that the loss function L is bounded by M . Suppose we have an online algorithm. Let h_i denote the classifier returned by the algorithm after observing (X_i, Y_i) . As before, the regret is defined as

$$R_T \sum_i L(h_i(X_i), Y_i) - \min_{h \in \mathcal{H}} \sum_{i=1}^T L(h(X_i), Y_i).$$

Let $R(h) = \mathbb{E}[L(h(X), Y)]$. First we bound the average risk.

Theorem 5 *With probability at least $1 - \delta$,*

$$\frac{1}{T} \sum_i R(h_i) \leq \frac{1}{T} \sum_i L(h_i(X_i), Y_i) + M \sqrt{\frac{2 \log(1/\delta)}{T}}.$$

Before proceeding let us recall Azuma's inequality. If V_i is a sequence of random variables that satisfy

$$\mathbb{E}[V_{i+1}|X_1, \dots, X_i] = 0$$

and $|V_i| \leq M$ then

$$P\left(\frac{1}{T} \sum_i V_i > \epsilon\right) \leq e^{-T\epsilon^2/(2M^2)}. \quad (2)$$

Proof. Let $V_i = R(h_i) - L(h_i(X_i), Y_i)$. Then

$$\mathbb{E}[V_i|X_1, \dots, X_{i-1}] = R(h_i) - \mathbb{E}[L(h_i(X_i), Y_i)|h_i] = R(h_i) - R(h_i) = 0.$$

Also $|V_i| \leq M$. Let

$$\epsilon = M \sqrt{(2/T) \log(1/\delta)}.$$

By (2),

$$P\left(\frac{1}{T} \sum_i X_i > \epsilon\right) \leq e^{-T\epsilon^2/(2M^2)} = \delta$$

and result follows from the definition of V_i . \square

We define our batch classifier as

$$h = \frac{1}{T} \sum_{i=1}^T h_i.$$

Theorem 6 *We have, with probability at least $1 - \delta$ that*

$$R(h) \leq \inf_{h \in \mathcal{H}} + \frac{R_T}{T} + 2M \sqrt{\frac{2 \log(1/\delta)}{T}}. \quad (3)$$

If we use the exponentially weighted algorithm then $R_T \leq \sqrt{T \log N / 2}$. Plugging this into (3) we have

$$R(h) \leq \inf_{h \in \mathcal{H}} + \sqrt{\frac{\log N}{2T}} + M \sqrt{\frac{2 \log(1/\delta)}{T}}.$$

Proof. By convexity,

$$L\left(\frac{1}{T} \sum_i h(X_i), Y_i\right) \leq \frac{1}{T} \sum_i L(h_i(X_i), Y_i).$$

By taking the expected value and using the fact that $h = \frac{1}{T} \sum_{i=1}^T h_i$,

$$R(h) \leq \frac{1}{T} \sum_i R(h_i).$$

From the previous theorem, with probability at least $1 - \delta/2$,

$$R(h) \leq \frac{1}{T} \sum_i L(h_i(X_i), Y_i) + M \sqrt{\frac{2 \log(2/\delta)}{T}}. \quad (4)$$

Since

$$R_T = \sum_i L(h(X_i), Y_i) - \min_h \in \mathcal{H} \sum_i L(h(X_i), Y_i)$$

(4) implies that

$$\begin{aligned} R(h) &\leq \frac{1}{T} \min_h \in \mathcal{H} \sum_i L(h(X_i), Y_i) + \frac{R_T}{T} + M \sqrt{\frac{2 \log(2/\delta)}{T}} \\ &= \frac{1}{T} \sum_i L(h_*(X_i), Y_i) + \frac{R_T}{T} + M \sqrt{\frac{2 \log(2/\delta)}{T}}. \end{aligned}$$

By Hoeffding's inequality, with probability at least $1 - \delta/2$,

$$\frac{1}{T} \sum_i L(h_*(X_i), Y_i) \leq R(h_*) + M \sqrt{\frac{2 \log(2/\delta)}{T}}.$$

Hence,

$$R(h) \leq R(h_*) + \frac{R_T}{T} + 2M \sqrt{\frac{2 \log(2/\delta)}{T}}.$$

□

Homework 1

Due Friday Feb 1 3:00 pm

1. Let $X_1, \dots, X_n \sim P$ where $X_i \in [0, 1]$ and P has density p . Let \hat{p} be the histogram estimator using m bins. Let $h = 1/m$. Recall that the L_2 error is $\int (\hat{p}(x) - p(x))^2 dx = \int \hat{p}^2(x)dx - 2 \int \hat{p}(x)p(x)dx + \int p^2(x)dx$. As usual, we may ignore the last term so we define the loss to be

$$L(h) = \int \hat{p}^2(x)dx - 2 \int \hat{p}(x)p(x)dx.$$

- (a) Suppose we used the direct estimator of the loss, namely, we replace the integral with the average to get

$$\widehat{L}(h) = \int \hat{p}^2(x)dx - \frac{2}{n} \sum_i \hat{p}(X_i).$$

Show that this fails in the sense that it is minimized by taking $h = 0$.

- (b) Recall that the leave-one-out estimator of the risk is

$$\widehat{L}(h) = \int \hat{p}^2(x)dx - \frac{2}{n} \sum_i \hat{p}_{(-i)}(X_i).$$

Show that

$$\widehat{L}(h) = \frac{2}{(n-1)h} - \frac{n+1}{n^2(n-1)h} \sum_j Z_j^2$$

where Z_j is the number of observations in bin j .

2. Let \hat{p}_h be the kernel density estimator (in one dimension) with bandwidth $h = h_n$. Let $s_n^2(x) = \text{var}(\hat{p}_h(x))$.

- (a) Show that, under appropriate conditions,

$$\frac{\hat{p}_h(x) - p_h(x)}{s_n(x)} \rightsquigarrow N(0, 1)$$

where $p_h(x) = \mathbb{E}[\hat{p}_h(x)]$.

Hint: Recall that the Lyapunov central limit theorem says the following: Suppose that Y_1, Y_2, \dots are independent. Let $\mu_i = \mathbb{E}[Y_i]$ and $\sigma_i^2 = \text{Var}(Y_i)$. Let $s_n^2 = \sum_{i=1}^n \sigma_i^2$. If

$$\lim_{n \rightarrow \infty} \frac{1}{s_n^{2+\delta}} \sum_{i=1}^n \mathbb{E}[|Y_i - \mu_i|^{2+\delta}] = 0$$

for some $\delta > 0$. Then $s_n^{-1} \sum_i (Y_i - \mu_i) \rightsquigarrow N(0, 1)$.

(b) Assume that the smoothness is $\beta = 2$. Suppose that the bandwidth h_n is chosen optimally. Show that

$$\frac{\widehat{p}_h(x) - p(x)}{s_n(x)} \rightsquigarrow N(b(x), 1)$$

for some constant $b(x)$ which is, in general, not 0.

3. Let $X_1, \dots, X_n \sim P$ where $X_i \in [0, 1]$. Assume that P has density p which has a bounded continuous derivative. Let $\widehat{p}_h(x)$ be the kernel density estimator. Show that, in general, the bias is of order $O(h)$ at the boundary. That is, show that $\mathbb{E}[\widehat{p}_h(0)] - p(0) = Ch$ for some $C > 0$.
4. Let p be a density on the real line. Assume that p is m -times continuously differentiable and that $\int |p^{(m)}|^2 < \infty$. Let K be a higher order kernel. This means that $\int K(y)dy = 1$, $\int y^j K(y)dy = 0$ for $1 \leq j \leq m-1$, $\int |y|^m K(y)dy < \infty$ and $\int K^2(y)dy < \infty$. Show that the kernel estimator with bandwidth h satisfies

$$\mathbb{E} \int (\widehat{p}(x) - p(x))^2 dx \leq C \left(\frac{1}{nh} + h^{2m} \right)$$

for some $C > 0$. What is the optimal bandwidth and what is the corresponding rate of convergence (using this bandwidth)?

5. Let $X_1, \dots, X_n \sim P$ where $X_i \in [0, 1]$ and P has density p . Let ϕ_1, ϕ_2, \dots be an orthonormal basis for $L_2[0, 1]$. Hence $\int_0^1 \phi_j^2(x)dx = 1$ for all j and $\int_0^1 \phi_j(x)\phi_k(x)dx = 0$ for $j \neq k$. Assume that the basis is uniformly bounded i.e. $\sup_j \sup_{0 \leq x \leq 1} |\phi_j(x)| \leq C < \infty$. We may expand p as $p(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$ where $\beta_j = \int \phi_j(x)p(x)dx$. Define

$$\widehat{p}(x) = \sum_{j=1}^k \widehat{\beta}_j \phi_j(x)$$

where $\widehat{\beta}_j = (1/n) \sum_{i=1}^n \phi_j(X_i)$.

- (a) Show that the risk is bounded by

$$\frac{ck}{n} + \sum_{j=k+1}^{\infty} \beta_j^2$$

for some constant $c > 0$.

- (b) Define the Sobolev ellipsoid $E(m, L)$ of order m as the set of densities of the form $p(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$ where $\sum_{j=1}^{\infty} \beta_j^2 j^{2m} < L^2$. Show that the risk for any density in $E(m, L)$ is bounded by $c[(k/n) + (1/k)^{2m}]$. Using this bound, find the optimal value of k and find the corresponding risk.

6. Recall that the total variation distance between two distributions P and Q is $\text{TV}(P, Q) = \sup_A |P(A) - Q(A)|$. In some sense, this would be the ideal loss function to use for density estimation. We only use L_2 because it is easier to deal with. Here you will explore some properties of TV.

- (a) Suppose that P and Q have densities p and q . Show that

$$\text{TV}(P, Q) = (1/2) \int |p(x) - q(x)| dx.$$

- (b) Let T be any mapping. Let X and Y be random variables. Then

$$\sup_A |P(T(X) \in A) - P(T(Y) \in A)| \leq \sup_A |P(X \in A) - P(Y \in A)|.$$

- (c) Let K be a kernel. Recall that the convolution of a density p with K is $(p \star K)(x) = \int p(z)K(x - z)dz$. Show that

$$\int |p \star K - q \star K| \leq \int |K| \int |p - q|.$$

Hence, smoothing reduces L_1 distance.

- (d) Let p be a density on \mathbb{R} and let p_n be a sequence of densities. Suppose that $\int (p - p_n)^2 \rightarrow 0$. Show that $\int |p - p_n| \rightarrow 0$.
- (e) Let \hat{p} be a histogram on \mathbb{R} with binwidth h . Under some regularity conditions it can be shown that

$$\mathbb{E} \int |\hat{p} - p| \approx \frac{\sqrt{2}}{\pi nh} \int \sqrt{p} + \frac{1}{4} h \int |p'|.$$

Hence, this risk can be unbounded if $\int \sqrt{p} = \infty$. A density is said to have a regularly varying tail of order r if $\lim_{x \rightarrow \infty} p(tx)/p(x) = t^r$ for all $t > 0$ and $\lim_{x \rightarrow -\infty} p(tx)/p(x) = t^r$ for all $t > 0$. Suppose that p has a regularly varying tail of order r with $r < -2$. Show that the risk bound above is bounded.

Homework 2
Due Friday Feb 22 3:00 pm
Submit a pdf file on Canvas

1. Consider data $(X_1, Y_1), \dots, (X_n, Y_n)$ where $X_i \in \mathbb{R}$ and $Y_i \in \mathbb{R}$. Inspired by the fact that $\mathbb{E}[Y|X = x] = \int yp(x, y)dy/p(x)$, define

$$\hat{m}(x) = \frac{\int y\hat{p}(x, y)dy}{\hat{p}(x)}$$

where

$$\hat{p}(x) = \frac{1}{n} \sum_i \frac{1}{h} K\left(\frac{X_i - x}{h}\right)$$

and

$$\hat{p}(x, y) = \frac{1}{n} \sum_i \frac{1}{h^2} K\left(\frac{X_i - x}{h}\right) K\left(\frac{Y_i - y}{h}\right).$$

Assume that $\int K(u)du = 1$ and $\int uK(u)du = 0$. Show that $\hat{m}(x)$ is exactly the kernel regression estimator that we defined in class.

2. Suppose that (X, Y) is bivariate Normal:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \eta \end{pmatrix}, \begin{pmatrix} \sigma^2 & \rho\sigma\tau \\ \rho\sigma\tau & \tau^2 \end{pmatrix}\right).$$

- (a) Show that $m(x) = \mathbb{E}[Y|X = x] = \alpha + \beta x$ and find explicit expressions for α and β .
(b) Find the maximum likelihood estimator $\hat{m}(x) = \hat{\alpha} + \hat{\beta}x$.
(c) Show that $|\hat{m}(x) - m(x)|^2 = O_P(n^{-1})$.

3. Let $m(x) = \mathbb{E}[Y|X = x]$. Let $X \in [0, 1]^d$, Divide $[0, 1]^d$ into cubes B_1, \dots, B_N whose sides have length h . The data are $(X_1, Y_1), \dots, (X_n, Y_n)$. In this problem, treat the X_i 's as fixed. Assume that the number of observations in each bin is positive. Let

$$\hat{m}(x) = \frac{1}{n(x)} \sum_i Y_i I(X_i \in B(x))$$

where $B(x)$ is the cube containing x and $n(x) = \sum_i I(X_i \in B(x))$. Assume that

$$|m(y) - m(x)| \leq L\|x - y\|_2$$

for all x, y . You may further assume that $\sup_x \text{Var}(Y|X = x) < \infty$.

- (a) Show that

$$|\mathbb{E}[\hat{m}(x)] - m(x)| \leq C_1 h$$

for some $C_1 > 0$. Also show that

$$\text{Var}(\hat{m}(x)) \leq \frac{C_2}{n(x)}$$

for some $C_2 > 0$.

(b) Now let X be random and assume that X has a uniform density on $[0, 1]^d$. Let $h \equiv h_n = (C \log n/n)^{1/d}$. Show that, for $C > 0$ large enough, $P(\min n_j = 0) \rightarrow 0$ as $n \rightarrow \infty$ where n_j is the number of observations in cube B_j .

4. Consider the RKHS problem

$$\hat{f} = \operatorname{argmin}_{f \in \mathcal{H}} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2, \quad (1)$$

for some Mercer kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. In this problem, you will prove that the above problem is equivalent to the finite dimensional one

$$\hat{\alpha} = \operatorname{argmin}_{\alpha \in \mathbb{R}^n} \|y - K\alpha\|_2^2 + \lambda \alpha^T K \alpha, \quad (2)$$

where $K \in \mathbb{R}^{n \times n}$ denotes the kernel matrix $K_{ij} = K(x_i, x_j)$.

Recall that the functions $K(\cdot, x_i)$, $i = 1, \dots, n$ are called the *representers of evaluation*. Recall that

- $\langle f, K(\cdot, x_i) \rangle_{\mathcal{H}} = f(x_i)$, for any function $f \in \mathcal{H}$
- $\|f\|_{\mathcal{H}}^2 = \sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j)$ for any function $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$.

(a) Let $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$, and consider defining a function $\tilde{f} = f + \rho$, where ρ is any function orthogonal to $K(\cdot, x_i)$, $i = 1, \dots, n$. Using the properties of the representers, prove that $\tilde{f}(x_i) = f(x_i)$ for all $i = 1, \dots, n$, and $\|\tilde{f}\|_{\mathcal{H}}^2 \geq \|f\|_{\mathcal{H}}^2$.

(b) Conclude from part (a) that in the infinite-dimensional problem (1), we need only consider functions of the form $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$, and that this in turn reduces to (2).

5. Let $X = (X(1), \dots, X(d)) \in \mathbb{R}^d$ and $Y \in \mathbb{R}$. In the questions below, make any reasonable assumptions that you need but state your assumptions.

- (a) Prove that $\mathbb{E}(Y - m(X))^2$ is minimized by choosing $m(x) = \mathbb{E}(Y|X = x)$.
- (b) Find the function $r(x)$ that minimizes $\mathbb{E}|Y - r(X)|$. (You can assume that the conditional cdf $F(y|X = x)$ is continuous and strictly increasing, for every x .)
- (c) Prove that $\mathbb{E}(Y - \beta^T X)^2$ is minimized by choosing $\beta_* = B^{-1}\alpha$ where $B = \mathbb{E}(XX^T)$ and $\alpha = (\alpha_1, \dots, \alpha_d)$ and $\alpha_j = \mathbb{E}(YX(j))$.

6. Consider the many Normal means problem where we observe $Y_i \sim N(\theta_i, 1)$ for $i = 1, \dots, d$. Let $\hat{\theta}$ minimize the penalized loss

$$\sum_i (Y_i - \theta_i)^2 + \lambda J(\theta).$$

Find an explicit expression for $\hat{\theta}$ in three cases: (i) $J(\theta) = ||\theta||_0$, (ii) $J(\theta) = ||\theta||_1$, (iii) $J(\theta) = ||\theta||_2$.

Homework 3
Due Friday March 29 3:00 pm
Submit a pdf file on Canvas

1. Get the iris data. (In R, use `data(iris)`.) There are 150 observations. The outcome is “Species” which has three values. The goal is to predict Species using the four covariates. Compare the following classifiers: (i) LDA, (ii) logistic regression, (iii) nearest neighbors. Note that you will need to figure out a way to deal with three classes when using logistic regression. Explain how you handled this. Summarize your results.
2. Use the iris data again but throw away the Species variable. Use k -means⁺⁺ clustering and mean-shift clustering. Compare the clusterings to the true group defined by Species. Which method worked better?
3. Download the data from <http://www-bcf.usc.edu/~gareth/ISL/Ch10Ex11.csv>. This is a gene expression dataset. There are 40 tissue samples with measurements on 1,000 genes. The first 20 data points are from healthy people. The second 20 data points are from diseased people.
 - (a) Use sparse logistic regression to classify the subject. (You may use the function `glmnet` in R if you like.) Explain how you chose λ . Summarize your findings.
 - (b) Now use a Sparse Additive Model as described in class. Summarize your findings.
 - (c) Now suppose we don’t know which are healthy and which are diseased. Apply clustering to put the data into two groups. Applying k -means clustering may not work well because the dimension is so high. Instead, you will need to do some sort of dimension reduction or sparse clustering. One very simple method is Sparse Alternate Similarity (arXiv:1602.07277). But you may use any method you like. Describe what you chose to do and what the results are.
4. Let $X = (X_1, \dots, X_d) \in \mathbb{R}^d$. Suppose that $X \sim N(\mu, \Sigma)$. Let $\Omega = \Sigma^{-1}$. Let $j \neq k$ be integers such that $1 \leq j < k \leq d$. Let $Z = (X_s : s \neq j, k)$.
 - (a) Show that the distribution of $(X_j, X_k)|Z$ is $N(a, B)$ and find a and B explicitly.
 - (b) Show that $X_j \perp\!\!\!\perp X_k|Z$ if and only if $\Omega_{jk} = 0$.
 - (c) Now let $X_1, \dots, X_n \sim N(\mu, \Sigma)$. Find the mle $\hat{\Omega}$.
5. Let $X = (X_1, X_2, X_3, X_4, X_5)$ be a random vector distributed as $X \sim N(0, \Sigma)$ where

$$\Sigma^{-1} = \begin{pmatrix} 3 & 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 1 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{pmatrix}.$$

- (a) What is the graph for X , viewed as an undirected graphical model?
- (b) List the maximal cliques of the graph.
- (c) Which of the following independence statements are true?
- $X_2 \perp\!\!\!\perp X_3 | X_1, X_2$
 - $X_3 \perp\!\!\!\perp X_4 | X_5$
 - $\{X_1, X_2\} \perp\!\!\!\perp X_3 | X_4, X_5$
 - $X_1 \perp\!\!\!\perp X_5 | X_3$
- (d) List the local Markov properties for this graphical model.
- (e) Simulate 100 observations from this model. Construct a graph using hypothesis testing. Report your results. Include your code.
6. Let $X = (X_1, \dots, X_4)$ where each variable is binary. Suppose the probability function is
- $$\log p(x) = \psi_\emptyset + \psi_1(x_1) + \psi_{12}(x_1, x_2) + \psi_{13}(x_1, x_3) + \psi_{24}(x_2, x_4) + \psi_{34}(x_3, x_4).$$
- (a) Draw the implied graph.
- (b) Write down all the independence and conditional independence relations implied by the graph.
- (c) Is the model graphical? Is the model hierarchical?
7. Let X_1, \dots, X_4 be binary. Draw the independence graphs corresponding to the following log-linear models (where $\alpha \in \mathbb{R}$). Also, identify whether each is graphical and/or hierarchical (or neither).
- $\log p(x) = \alpha + 11x_1 + 2x_2 + 3x_3$
 - $\log p(x) = \alpha + 2x_2 + 1.5x_3 + 17x_4 + 12x_2x_3 + 78x_2x_4 + 3x_3x_4 + 32x_2x_3x_4$
 - $\log p(x) = \alpha + 9x_1 + 2x_2 + 1.5x_3 + 17x_4 + 12x_2x_3 + 3x_3x_4 + x_1x_4 + 2x_1x_2$
 - $\log p(x) = \alpha + 115x_1x_2x_3x_4.$
8. Consider the log-linear model

$$\log p(x) = \beta_0 + x_1x_2 + x_2x_3 + x_3x_4.$$

Simulate $n = 1000$ random vectors from this distribution. (Show your code.) Fit the model

$$\log p(x) = \beta_0 + \sum_j \beta_j x_j + \sum_{k < \ell} \beta_{k\ell} x_k x_\ell$$

using maximum likelihood. Report your estimators. Use hypothesis testing to decide which parameters are non-zero. Compare the selected model to the true model.

9. Let $X_1, \dots, X_n \in \mathbb{R}^d$. Let Σ be the $d \times d$ covariance matrix for X_i . The covariance graph G puts an edge between (j, k) if $\Sigma_{jk} \neq 0$. Here we will use the bootstrap to estimate the covariance graph.

Let Σ have the following form: $\Sigma_{jj} = 1$, $\Sigma_{j,k} = a$ if $|j - k| = 1$ and $\Sigma_{j,k} = 0$ otherwise. Here, $a = 1/4$.

Let $d = 100$ and $n = 50$. Generate n observations. Compute a 95 percent bootstrap confidence set for Σ using the bootstrap distribution

$$\mathbb{P}\left(\max_{j,k} \sqrt{n}|\widehat{\Sigma}_{jk}^* - \widehat{\Sigma}_{jk}| \leq t \mid X_1, \dots, X_n\right).$$

This gives (uniform) confidence intervals for all the elements of Σ_{jk} . For each (j, k) , put an edge if the confidence interval for Σ_{jk} excludes 0. Plot your graph. Try this for different values of a . Summarize your results.

10. Let $A \in \{0, 1\}$ be a binary treatment variable and let $Y \in \mathbb{R}$ be the response variable. Let $(Y(0), Y(1))$ be the counterfactual variables where $Y = AY(1) + (1 - A)Y(0)$. Assume that

$$Y = \alpha + \gamma A + \sum_{j=1}^d \beta_j X_j + \epsilon$$

where (X_1, \dots, X_d) are confounding variables and $\mathbb{E}[\epsilon | X_1, \dots, X_d] = 0$. Assume there are no unmeasured variables.

- (a) Let $\theta = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)]$. Show that $\theta = \gamma$.
- (b) Suppose now that we do not observe the confounding variables X_j . All we observe is $(A_1, Y_1), \dots, (A_n, Y_n)$. Suppose, unaware of the confounding variables, we fit the linear model $Y = \alpha + \rho A + \delta$ where $\mathbb{E}[\delta] = 0$. Let $\widehat{\rho}$ be the least squares estimator. Show that $\widehat{\rho} \xrightarrow{P} \gamma + \Delta$ for some Δ . Find an explicit expression for Δ .

11. Consider a sequence of time ordered random variables

$$X_1, A_1, Y_1, X_2, A_2, Y_2, X_3, A_3, Y_3, \dots, X_T, A_T, Y_T.$$

Here, the X_j 's are covariates, the A_j 's are binary treatment variables and the Y_j 's are the response of interest. Assume there are no unobserved confounding variables. The DAG for this model has all directed arrows from the past into the future. That is, the parents for each variables are all variables in its past. For example, the parent of A_1 is X_1 . The parents of Y_1 are (X_1, A_1) . The parents of X_2 are (X_1, A_1, Y_1) and so on. Let p denote the joint density of all these variables.

- (a) Find an explicit expression (in terms of p) for

$$E[Y_T | A_1 = a_1, \dots, A_T = a_T].$$

- (b) Find an explicit expression (in terms of p) for

$$E[Y_T | \text{set } (A_1 = a_1, A_2 = a_2, \dots, A_T = a_T)].$$

Homework 4
Due Friday April 19 3:00 pm
Submit a pdf file on Canvas

1. Consider the directed graph with vertices $V = \{X_1, X_2, X_3, X_4, X_5\}$ and edge set $E = \{(1, 3), (2, 3), (3, 4), (3, 5)\}$.
 - (a) List all the independence statements implied by this graph.
 - (b) Find the causal distribution $p(x_4 | \text{set } x_3 = s)$.
 - (c) Find the implied undirected graph for these random variables. Which independence statements get lost in the undirected graph (if any)?
2. Let $d \geq 2$, and let $X_1, \dots, X_n \sim P$ where $X_i = (X_i(1), \dots, X_i(d)) \in \mathbb{R}^d$. Assume that the coordinates of X_i are independent. Further, assume that $X_i(j) \sim \text{Bernoulli}(p_j)$ where $0 < c \leq p_j \leq C < 1$. Let \mathcal{P} be all such distributions. Let

$$R_n = \inf_{\hat{p}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \|\hat{p} - p\|_\infty.$$

Find lower and upper bounds on the minimax risk.

3. Let $\{p_\theta : \theta \in \Theta\}$ where $\Theta \subset \mathbb{R}$ be a parametric model. Suppose that the model satisfies the usual regularity conditions. In particular, the Fisher information $I(\theta)$ is positive and smooth and the mle has the usual nice properties. Let the loss function be $L(\hat{\theta}, \theta) = H(p_{\hat{\theta}}, p_\theta)$ where H denotes Hellinger distance. Find the minimax rate.
4. Let $Y = (Y_1, \dots, Y_d) \sim N(\theta, I)$ where $\theta = (\theta_1, \dots, \theta_d)$. Assume that $\theta \in \Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq 1\}$. Let

$$R_d = \inf_{\hat{\theta}} \sup_{\theta \in \Theta} \mathbb{E}_\theta \|\hat{\theta} - \theta\|^2.$$

Show that $c \log d \leq R_d \leq C \log d$ for some constants c and C .

5. Let $X_1, \dots, X_n \sim F$ where F is some distribution on \mathbb{R} . Suppose we put a Dirichlet process prior on F :

$$F \sim \text{DP}(\alpha, F_0).$$

- (a) Recall the stick-breaking construction. Show that $\mathbb{E}(\sum_{j=1}^\infty W_j) = 1$.
- (b) Simulate $n = 10$ data points from a $N(0, 1)$. Try three values of α : namely, $\alpha = .1$, $\alpha = 1$ and $\alpha = 10$. Compute the 95 percent Bayesian confidence band and the 95 percent DKW band. Plot the results for one example. Now repeat the simulation 1,000 times and report the coverage probability for each confidence band.
6. For $i = 1, \dots, n$ and $j = 1, 2, \dots$ let

$$X_{ij} = \theta_j + \epsilon_{ij}$$

where all the ϵ'_{ij} s are independent $N(0,1)$. The parameter is $\theta = (\theta_1, \theta_2, \dots, \theta_n)$. Assume that $\sum_j \theta_j^2 < \infty$. Due to sufficiency, we can reduce the problem to the sample means. Thus let $Y_j = n^{-1} \sum_{i=1}^n X_{ij}$. So the model is $Y_j \sim N(\theta_j, 1/n)$ for $j = 1, 2, 3, \dots$. We will put a prior π on θ as follows. We take each θ_j to be independent and we take $\theta_j \sim N(0, \tau_j^2)$.

- (a) Find the posterior for θ . Find the posterior mean $\hat{\theta}$.
- (b) Suppose that $\sum_j \tau_j^2 < \infty$. Show that $\hat{\theta}$ is consistent, that is, $\|\hat{\theta} - \theta\|^2 \xrightarrow{P} 0$.
- (c) Now suppose that θ is in the Sobolev ball

$$\Theta = \left\{ \theta = (\theta_1, \theta_2, \dots) : \sum_j j^{2p} \theta_j^2 \leq C^2 \right\}$$

where $p > 1/2$. The minimax (for squared error loss) for this problem is $R_n \asymp n^{-2p/(2p+1)}$. Let $\tau_j^2 = (1/j)^{2r}$. Find r so that the posterior mean achieves the minimax rate.

36-708 Statistical Methods for Machine Learning

Homework #1 Solutions

February 1, 2019

Problem 1 [15 pts.]

Let $X_1, \dots, X_n \sim P$ where $X_i \in [0, 1]$ and P has density p . Let \widehat{p} be the histogram estimator using m bins. Let $h = 1/m$. Recall that the L_2 error is $\int (\widehat{p}(x) - p(x))^2 dx = \int \widehat{p}^2(x)dx - 2 \int \widehat{p}(x)p(x)dx + \int p^2(x)dx$. As usual, we may ignore the last term so we define the loss to be

$$L(h) = \int \widehat{p}^2(x)dx - 2 \int \widehat{p}(x)p(x)dx.$$

- (a) Suppose we used the direct estimator of the loss, namely, we replace the integral with the average to get

$$\widehat{L}(h) = \int \widehat{p}^2(x)dx - \frac{2}{n} \sum_i \widehat{p}(X_i).$$

Show that this fails in the sense that it is minimized by taking $h = 0$.

- (b) Recall that the leave-one-out estimator of the risk is

$$\widehat{L}(h) = \int \widehat{p}^2(x)dx - \frac{2}{n} \sum_i \widehat{p}_{-(i)}(X_i),$$

Show that

$$\widehat{L}(h) = \frac{2}{(n-1)h} - \frac{n+1}{n^2(n-1)h} \sum_j Z_j^2$$

where Z_j is the number of observations in bin j .

Solution.

Define

$$\widehat{\theta}_j = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(X_i \in B_j) \quad \text{and} \quad Z_j = n\widehat{\theta}_j$$

for $j = 1, \dots, m$.

(a) (7 pts.)

$$\begin{aligned}
 \widehat{L}(h) &= \int_0^1 \widehat{p}^2(x) dx - \frac{2}{n} \sum_i \widehat{p}(X_i) \\
 &= \int_0^1 \left(\sum_{j=1}^m \frac{\widehat{\theta}_j}{h} \mathbb{1}(x \in B_j) \right)^2 dx - \frac{2}{n} \sum_{i=1}^n \sum_{j=1}^m \frac{\widehat{\theta}_j}{h} \mathbb{1}(X_i \in B_j) \\
 &= \frac{1}{h^2} \int_0^1 \left(\sum_{k=1}^m \sum_{j=1}^m \widehat{\theta}_j \widehat{\theta}_k \mathbb{1}(x \in B_j \cap B_k) \right) dx - \frac{2}{h} \sum_{j=1}^m \widehat{\theta}_j \cdot \frac{1}{n} \sum_{i=1}^n \mathbb{1}(X_i \in B_j) \\
 &= \frac{1}{h^2} \int_0^1 \left(\sum_{j=1}^m \widehat{\theta}_j^2 \mathbb{1}(x \in B_j) \right) dx - \frac{2}{h} \sum_{j=1}^m \widehat{\theta}_j^2 \\
 &= \frac{1}{h^2} \sum_{j=1}^m \widehat{\theta}_j^2 \int_0^1 \mathbb{1}(x \in B_j) dx - \frac{2}{h} \sum_{j=1}^m \widehat{\theta}_j^2 \\
 &= \frac{1}{h} \sum_{j=1}^m \widehat{\theta}_j^2 - \frac{2}{h} \sum_{j=1}^m \widehat{\theta}_j^2 \\
 &= -\frac{1}{h} \sum_{j=1}^m \widehat{\theta}_j^2 \\
 &= -\frac{1}{hn^2} \sum_{j=1}^m Z_j^2
 \end{aligned}$$

Considering the last quantity , we have that:

$$\begin{aligned}
 \sum_{j=1}^m Z_j^2 &\geq \sum_{j=1}^m Z_j = n \quad , \quad \sum_{j=1}^m Z_j^2 = \sum_{j=1}^m Z_j Z_j \leq \sum_{j=1}^m Z_j n = n^2 \\
 \implies -\frac{1}{h} &\leq \widehat{L}(h) \leq -\frac{1}{nh}
 \end{aligned}$$

So $\widehat{L}(h) \rightarrow -\infty$ as $h \rightarrow 0$. Therefore, this loss is minimized by taking $h = 0$.

(b) (8 pts.)

From part (a) we have

$$\int \widehat{p}^2(x) dx = \frac{1}{h} \sum_{j=1}^m \widehat{\theta}_j^2. \tag{1}$$

And the second term in the leave-one-out loss is

$$\begin{aligned}
 \frac{2}{n} \sum_{i=1}^n \widehat{p}_{(-i)}(X_i) &= \frac{2}{n(n-1)h} \sum_{j=1}^m \sum_{i=1}^n \mathbb{1}(X_i \in B_j) \sum_{k \neq i} \mathbb{1}(X_k \in B_j) \\
 &= \frac{2}{n(n-1)h} \sum_{j=1}^m \sum_{i=1}^n \mathbb{1}(X_i \in B_j) (n\widehat{\theta}_j - \mathbb{1}(X_i \in B_j)) \\
 &= \frac{2}{n(n-1)h} \sum_{j=1}^m (n^2 \widehat{\theta}_j^2 - n\widehat{\theta}_j).
 \end{aligned} \tag{2}$$

Taking the difference of (1) and (2), we get

$$\begin{aligned}
 \widehat{L}(h) &= \frac{1}{h} \sum_{j=1}^m \widehat{\theta}_j^2 - \frac{2}{n(n-1)h} \sum_{j=1}^m (n^2 \widehat{\theta}_j^2 - n \widehat{\theta}_j) \\
 &= \frac{2}{(n-1)h} \underbrace{\sum_{j=1}^m \widehat{\theta}_j}_{=1} + \sum_{j=1}^m \widehat{\theta}_j^2 \left(\frac{1}{h} - \frac{2n}{(n-1)h} \right) \\
 &= \frac{2}{(n-1)h} - \frac{n+1}{(n-1)h} \sum_{j=1}^m \widehat{\theta}_j^2 \\
 &= \frac{2}{(n-1)h} - \frac{n+1}{n^2(n-1)h} \sum_{j=1}^m Z_j^2.
 \end{aligned}$$

Problem 2 [15 pts.]

Let \widehat{p}_h be the kernel density estimator (in one dimension) with bandwidth $h = h_n$. Let $s_n^2(x) = \text{Var}(\widehat{p}_h(x))$.

(a) Show that

$$\frac{\widehat{p}_h(x) - p(x)}{s_n(x)} \rightsquigarrow N(0, 1)$$

where $p_h(x) = \mathbb{E}[\widehat{p}_h(x)]$.

Hint: Recall that the Lyapunov central limit theorem says the following: Suppose that Y_1, Y_2, \dots are independent. Let $\mu_i = \mathbb{E}[Y_i]$ and $\sigma_i^2 = \text{Var}(Y_i)$. Let $s_n^2 = \sum_{i=1}^n \sigma_i^2$. If

$$\lim_{n \rightarrow \infty} \frac{1}{s_n^{2+\delta}} \sum_{i=1}^n \mathbb{E}[|Y_i - \mu_i|^{2+\delta}] = 0$$

for some $\delta > 0$. Then $s_n^{-1} \sum_i (Y_i - \mu_i) \rightsquigarrow N(0, 1)$.

(b) Assume that the smoothness is $\beta = 2$. Suppose that the bandwidth h_n is chosen optimally. Show that

$$\frac{\widehat{p}_h(x) - p(x)}{s_n(x)} \rightsquigarrow N(b(x), 1)$$

for some constant $b(x)$ which is, in general, not 0.

Solution.

(a) [8 pts.]

Caveat: The classical Central Limit Theorem cannot be applied here, as $h = h_n$ is a function of n and thus the $K\left(\frac{\|x-X_i\|}{h}\right)$ are not identically distributed. However, as the hint suggests, the Lyapunov CLT still holds for non-identically distributed random variables.

Claim. Let $p > 1$. Then

$$\mathbb{E}\left[\left|\frac{1}{h} K\left(\frac{\|x-X_i\|}{h}\right) - p_h(x)\right|^p\right] = \Theta\left(\frac{1}{h^{p-1}}\right).$$

Proof. See appendix

Now

$$\begin{aligned}\mathbb{E}\left[\left|\frac{1}{nh}K\left(\frac{\|x-X_i\|}{h}\right)-\frac{p_h(x)}{n}\right|^{2+\delta}\right] &= \frac{1}{n^{2+\delta}}\mathbb{E}\left[\left|\frac{1}{h}K\left(\frac{\|x-X_i\|}{h}\right)-p_h(x)\right|^{2+\delta}\right] \\ &= \Theta\left(\frac{1}{n^{2+\delta}h^{1+\delta}}\right),\end{aligned}$$

and

$$\begin{aligned}s_n^2 &= \sum_{i=1}^n \mathbb{E}\left[\left|\frac{1}{nh}K\left(\frac{\|x-X_i\|}{h}\right)-\frac{p_h(x)}{n}\right|^2\right] \\ &= \frac{1}{n}\mathbb{E}\left[\left|\frac{1}{h}K\left(\frac{\|x-X_i\|}{h}\right)-p_h(x)\right|^2\right] \\ &= \Theta\left(\frac{1}{nh}\right).\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{1}{s_n^{2+\delta}} \sum_{i=1}^n \mathbb{E}\left[\left|\frac{1}{nh}K\left(\frac{\|x-X_i\|}{h}\right)-\frac{p_h(x)}{n}\right|^{2+\delta}\right] &= \Theta((nh)^{1+\frac{\delta}{2}}) \cdot n \cdot \Theta\left(\frac{1}{n^{2+\delta}h^{1+\delta}}\right) \\ &= \Theta((nh)^{-\frac{\delta}{2}}) \\ &\rightarrow 0,\end{aligned}$$

as $n \rightarrow \infty$ and $nh \rightarrow \infty$, for any $\delta > 0$. So, by the Lyapunov CLT,

$$\frac{\widehat{p}_h(x) - p_h(x)}{s_n(x)} \rightsquigarrow N(0, 1).$$

(b) [7 pts.] First note

$$\begin{aligned}\frac{\widehat{p}_h(x) - p(x)}{s_n(x)} &= \frac{\widehat{p}_h(x) - p_h(x)}{s_n(x)} + \frac{p_h(x) - p(x)}{s_n(x)} \\ &= \frac{\widehat{p}_h(x) - p_h(x)}{s_n(x)} + \frac{\text{Bias}(p_h(x))}{\sqrt{\text{Var}(\widehat{p}_h(x))}}.\end{aligned}$$

From Theorem 5, the optimal bandwidth is $h_n = \Theta(n^{-1/5})$.

Now from part (a), we have

$$\text{Var}(\widehat{p}_h(x)) = \Theta\left(\frac{1}{nh}\right)$$

and from Lemma 3,

$$\text{Bias}(p_h(x)) = O(h^2).$$

Therefore,

$$\begin{aligned}
 \frac{\widehat{p}_h(x) - p(x)}{s_n(x)} &= \frac{\widehat{p}_h(x) - p_h(x)}{s_n(x)} + \frac{\text{Bias}(p_h(x))}{\sqrt{\text{Var}(\widehat{p}_h(x))}} \\
 &= \frac{\widehat{p}_h(x) - p_h(x)}{s_n(x)} + \frac{O(h^2)}{\Theta\left(\frac{1}{(nh)^{1/2}}\right)} \\
 &= \frac{\widehat{p}_h(x) - p_h(x)}{s_n(x)} + \frac{O(n^{-2/5})}{\Theta\left(n^{-2/5}\right)} \\
 &= \underbrace{\frac{\widehat{p}_h(x) - p_h(x)}{s_n(x)}}_{\sim N(0,1)} + O(1) \\
 &\rightsquigarrow N(b(x), 1).
 \end{aligned}$$

Problem 3 [10 pts.]

Let $X_1, \dots, X_n \sim P$ where $X_i \in [0, 1]$. Assume that P has density p which has bounded continuous derivative. Let $\widehat{p}_h(x)$ be the kernel density estimator. Show that, in general, the bias is of order $O(h)$ at the boundary. That is, show that $\mathbb{E}[\widehat{p}_h(0)] - p(0) = Ch$ for some $C > 0$.

Solution.

$$\begin{aligned}
 \mathbb{E}[\widehat{p}(0)] &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{-X_i}{h}\right)\right] \\
 &= \mathbb{E}\left[\frac{1}{h} K\left(\frac{X_i}{h}\right)\right] \\
 &= \frac{1}{h} \int_0^1 K\left(\frac{u}{h}\right) p(u) du \\
 &= \int_0^{1/h} K(t) p(ht) dt \\
 &= \int_0^{1/h} K(t) \left(p(0) + ht \cdot \partial_+ p(0) + \frac{h^2 t^2}{2} \cdot \partial_+^2 p(0) + o(h^2)\right) dt \quad \text{let } t = \frac{u}{h} \\
 &= p(0) \int_0^{1/h} K(t) dt + O(h) \int_0^{1/h} t K(t) dt + O(h^2) \underbrace{\int_0^{1/h} t^2 K(t) dt}_{\leq \sigma_K^2 / 2 < \infty} \\
 &\leq p(0) + O(h),
 \end{aligned}$$

where we assumed $K(\cdot)$ is supported on $[-1, 1]$, $h \leq 1$, and $\int_0^{1/h} t K(t) dt$ is bounded.

Problem 4 [10 pts.]

Let p be a density on the real line. Assume that p is m -times continuously differentiable and that $\int |p^{(m)}|^2 < \infty$. Let K be a higher order kernel. This means that $\int K(y)dy = 1$, $\int y^j K(y)dy = 0$ for $1 \leq j \leq m-1$, $\int |y|^m K(y)dy < \infty$ and $\int K^2(y)dy < \infty$. Show that the kernel estimator with bandwidth h satisfies

$$\mathbb{E} \int (\hat{p}(x) - p(x))^2 dx \leq C \left(\frac{1}{nh} + h^{2m} \right)$$

for some $C > 0$. What is the optimal bandwidth and what is the corresponding rate of convergence (using this bandwidth)?

Solution.

We assume p has bounded m derivatives, and so $p \in \Sigma(m, L)$ for some constant $L > 0 \in \mathbb{R}$. Let's first analyze the bias $b(x)$:

$$\begin{aligned} \mathbb{E}[\hat{p}(x)] - p(x) &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x-x_i}{h}\right)\right] - p(x) \\ &= \mathbb{E}\left[\frac{1}{h} K\left(\frac{x-x_1}{h}\right)\right] - p(x) \\ &= \int \frac{1}{h} K\left(\frac{x-u}{h}\right) p(u) du - p(x) \\ &= \int K(t) p(x-th) dt - p(x) \quad \text{where } t = \frac{x-u}{h} \\ &= \int K(t) \left[p(x) - thp'(x) + \frac{t^2 h^2}{2} p''(x) + \dots + \frac{(-th)^{m-1}}{(m-1)!} p^{(m-1)}(x) \right. \\ &\quad \left. + \frac{(-th)^m}{m!} p^{(m)}(w) \right] dt - p(x) \quad w \in (x-th, x), \text{ Taylor Exp.} \end{aligned}$$

Given $\int K(y)dy = 1$, then $\int K(t)p(x)dt = p(x)$ and $\int y^j K(y)dy = 0$ for $1 \leq j \leq m-1$, so we are left with:

$$\begin{aligned} |\mathbb{E}[\hat{p}(x)] - p(x)| &= \left| \int K(t) \frac{(-th)^m}{(m)!} p^{(m)}(w) dt \right| \\ &\leq \frac{Lh^m}{m!} \left| \int K(t) t^m dt \right| \\ &\leq \frac{Lh^m}{m!} \int |K(t)| |t|^m dt = Ch^m \quad \text{for some } 0 < C < \infty \end{aligned}$$

And so we have that $\int b(x)^2 dx \leq C'h^{2m}$ for some $0 < C' < \infty$.

Analyzing now the variance we have that:

$$\begin{aligned}
 \mathbb{V}(\hat{p}(x)) &= \mathbb{V}\left(\frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x-x_i}{h}\right)\right) \\
 &= \frac{1}{nh^2} \mathbb{V}\left(K\left(\frac{x-x_1}{h}\right)\right) \\
 &\leq \frac{1}{nh^2} \mathbb{E}\left(K\left(\frac{x-x_1}{h}\right)^2\right) \\
 &= \frac{1}{nh^2} \int K\left(\frac{x-x_1}{h}\right)^2 p(x) dx \\
 &= \frac{1}{nh} \int K(t)^2 p(x-th) dt \quad \text{where } t = \frac{x-u}{h} \\
 &\leq \frac{\sup_x p(x)}{nh} \int K(t)^2 dt \leq \frac{C''}{nh} \quad \text{for some } 0 < C'' < \infty
 \end{aligned}$$

Since densities in $\Sigma(m, L)$ are uniformly bounded.

The optimal bandwidth is therefore:

$$\frac{\partial \left(\frac{1}{nh} + h^{2m} \right)}{\partial h} = 0 \implies -\frac{1}{nh^2} + 2mh^{2m-1} = 0 \implies h^* = (2mn)^{-\frac{1}{2m+1}} \asymp n^{-\frac{1}{2m+1}}$$

And so the convergence rate is:

$$\mathbb{E}\left[\int (\hat{p}(x) - p(x))^2 dx\right] \leq n^{-\frac{2m}{2m+1}}$$

Problem 5 [15 pts.]

Let $X_1, \dots, X_n \sim P$ where $X_i \in [0, 1]$ and P has density p . Let ϕ_1, ϕ_2, \dots be an orthonormal basis for $L_2[0, 1]$. Hence $\int_0^1 \phi_j^2(x) dx = 1$ for all j and $\int_0^1 \phi_j(x) \phi_k(x) dx = 0$ for $j \neq k$. Assume that the basis is uniformly bounded, i.e. $\sup_j \sup_{0 \leq x \leq 1} |\phi_j(x)| \leq C < \infty$. We may expand p as $p(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$ where $\beta_j = \int \phi_j(x) p(x) dx$. Define

$$\widehat{p}(x) = \sum_{j=1}^k \widehat{\beta}_j \phi_j(x)$$

where $\widehat{\beta}_j = (1/n) \sum_{i=1}^n \phi_j(X_i)$.

- (a) Show that the risk is bounded by

$$\frac{ck}{n} + \sum_{j=k+1}^{\infty} \beta_j^2$$

for some constant $c > 0$.

- (b) Define the Sobolev ellipsoid $E(m, L)$ of order m as the set of densities of the form $p(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$ where $\sum_{j=1}^{\infty} \beta_j^2 j^{2m} < L^2$. Show that the risk for any density in $E(m, L)$ is bounded by $c[(k/n) + (1/k)^{2m}]$. Using this bound, find the optimal value of k and find the corresponding risk.

Solution.

(a) (10 pts.)

First note,

$$\begin{aligned}\mathbb{E}[\widehat{\beta}_j] &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n \phi_j(X_i)\right] \\ &= \mathbb{E}[\phi_j(x)] \\ &= \int_0^1 p(x) \phi_j(x) dx \\ &= \beta_j.\end{aligned}$$

So $\widehat{\beta}_j$ is unbiased. Now,

$$\begin{aligned}R(\widehat{p}(x)) &= \mathbb{E}\left[\int (\widehat{p}(x) - p(x))^2 dx\right] \\ &= \mathbb{E}\left[\int \left(\sum_{j=1}^k \widehat{\beta}_j \phi_j(x) - \sum_{j=1}^{\infty} \beta_j \phi_j(x)\right)^2 dx\right] \\ &= \mathbb{E}\left[\int \left(\sum_{j=1}^k (\widehat{\beta}_j - \beta_j) \phi_j(x) - \sum_{j=k+1}^{\infty} \beta_j \phi_j(x)\right)^2 dx\right] \\ &= \mathbb{E}\left[\sum_{j=1}^k (\widehat{\beta}_j - \beta_j)^2 + \sum_{j=k+1}^{\infty} \beta_j^2\right] \quad \text{since } \int \phi_i \phi_j = \delta_{ij} \\ &= \sum_{j=1}^k \text{Var}(\widehat{\beta}_j) + \sum_{j=k+1}^{\infty} \beta_j^2 \\ &= \frac{k}{n} \text{Var}(\phi_j(X_i)) + \sum_{j=k+1}^{\infty} \beta_j^2 \\ &\leq \frac{C^2 k}{n} + \sum_{j=k+1}^{\infty} \beta_j^2.\end{aligned}$$

(b) (5 pts.)

$$\begin{aligned}\sup_{p \in E(m, L)} R(\widehat{p}(x)) &\leq \frac{C^2 k}{n} + \sum_{j=k+1}^{\infty} \beta_j^2 \quad \text{from part (a)} \\ &= \frac{C^2 k}{n} + \frac{k^{2m} \sum_{j=k+1}^{\infty} \beta_j^2}{k^{2m}} \\ &\leq \frac{C^2 k}{n} + \frac{\sum_{j=k+1}^{\infty} \beta_j^2 j^{2m}}{k^{2m}} \\ &\leq \frac{C^2 k}{n} + \frac{L^2}{k^{2m}} \\ &\leq \max\{C^2, L^2\} \left(\frac{k}{n} + \frac{1}{k^{2m}} \right)\end{aligned}$$

Optimal k (up to some constant) can be found by, $\frac{k}{n} = \frac{1}{k^{2m}}$, which is, $k = O(n^{1/(2m+1)})$. And the corresponding risk is of the rate, $O(n^{-2m/(2m+1)})$.

Problem 6 [35 pts.]

Recall that the total variation distance between two distributions P and Q is $\text{TV}(P, Q) = \sup_A |P(A) - Q(A)|$. In some sense, this would be the ideal loss function to use for density estimation. We only use L_2 because it is easier to deal with. Here you will explore some properties of TV.

- (a) Suppose that P and Q have densities p and q . Show that

$$\text{TV}(P, Q) = (1/2) \int |p(x) - q(x)| dx.$$

- (b) Let T be any mapping. Let X and Y be random variables. Then

$$\sup_A |P(T(X) \in A) - P(T(Y) \in A)| \leq \sup_A |P(X \in A) - P(Y \in A)|.$$

- (c) Let K be a kernel. Recall that the convolution of a density p with K is $(p * K)(x) = \int p(z)K(x - z) dz$. Show that

$$\int |p * K - q * K| \leq \int |K| \int |p - q|.$$

Hence, smoothing reduces L_1 distance.

- (d) Let p be a density on \mathbb{R} and let p_n be a sequence of densities. Suppose that $\int (p - p_n)^2 \rightarrow 0$. Show that $\int |p - p_n| \rightarrow 0$.

- (e) Let \hat{p} be a histogram on \mathbb{R} with binwidth h . Under some regularity conditions it can be shown that

$$\mathbb{E} \int |p - p_n| \approx \frac{\sqrt{2}}{\pi nh} \int \sqrt{p} + \frac{1}{4}h \int |p'|.$$

Hence, this risk can be unbounded if $\int \sqrt{p} = \infty$. A density is said to have a regularly varying tail of order r if $\lim_{x \rightarrow \infty} p(tx)/p(x) = t^r$ for all $t > 0$ and $\lim_{x \rightarrow -\infty} p(tx)/p(x) = t^r$ for all $t > 0$. Suppose that p has a regularly varying tail of order r with $r < -2$. Show that the risk bound above is bounded.

Solution.

- (a) **(10 pts.)**

For any measurable $B \subseteq \mathbb{R}$,

$$\begin{aligned} \frac{1}{2} \int |p - q| &= \frac{1}{2} \int |p(x) - q(x)| dx \\ &\geq \frac{1}{2} \int_B (p(x) - q(x)) dx + \frac{1}{2} \int_{\mathbb{R} \setminus B} (q(x) - p(x)) dx \\ &= \frac{1}{2} \int_B p(x) dx - \frac{1}{2} \int_B q(x) dx + \frac{1}{2} \int_{\mathbb{R} \setminus B} q(x) dx - \frac{1}{2} \int_{\mathbb{R} \setminus B} p(x) dx \\ &= \frac{1}{2} \int_B p(x) dx - \frac{1}{2} \int_B q(x) dx + \frac{1}{2} \left(1 - \int_B q(x) dx\right) - \frac{1}{2} \left(1 - \int_B p(x) dx\right) \\ &= \left(\int_B p(x) dx - \int_B q(x) dx \right) \\ &= P(B) - Q(B) \end{aligned}$$

$$\implies \frac{1}{2} \int |p - q| \geq P(B) - Q(B) \text{ for any measurable } B \subseteq \mathbb{R}.$$

By noting,

$$\frac{1}{2} \int |p - q| = \frac{1}{2} \int |q - p|,$$

parallel reasoning shows

$$\frac{1}{2} \int |p - q| \geq Q(B) - P(B) \text{ for any measurable } B \subseteq \mathbb{R}.$$

So together we have,

$$\frac{1}{2} \int |p - q| \geq |P(B) - Q(B)|$$

and thus

$$\frac{1}{2} \int |p - q| \geq \sup_{B \subseteq \mathbb{R}} |P(B) - Q(B)|, \quad (3)$$

for any measurable $B \subseteq \mathbb{R}$.

Now consider the set

$$B' = \{x \in \mathbb{R} : p(x) > q(x)\}.$$

B' is measurable and

$$\begin{aligned} \frac{1}{2} \int |p - q| &= \frac{1}{2} \int |p(x) - q(x)| dx \\ &= \frac{1}{2} \int_{B'} (p(x) - q(x)) dx + \frac{1}{2} \int_{\mathbb{R} \setminus B'} (q(x) - p(x)) dx \\ &= \frac{1}{2} \int_{B'} p(x) dx - \frac{1}{2} \int_{B'} q(x) dx + \frac{1}{2} \int_{\mathbb{R} \setminus B'} q(x) dx - \frac{1}{2} \int_{\mathbb{R} \setminus B'} p(x) dx \\ &= \frac{1}{2} \int_{B'} p(x) dx - \frac{1}{2} \int_{B'} q(x) dx + \frac{1}{2} \left(1 - \int_{B'} q(x) dx\right) - \frac{1}{2} \left(1 - \int_{B'} p(x) dx\right) \\ &= \left(\int_{B'} p(x) dx - \int_{B'} q(x) dx \right) \\ &= P(B') - Q(B') \\ &= |P(B') - Q(B')|. \end{aligned}$$

We have found a set $B' \subseteq \mathbb{R}$ such that

$$\frac{1}{2} \int |p - q| = |P(B') - Q(B')|,$$

therefore,

$$\frac{1}{2} \int |p - q| \leq \sup_{B \subseteq \mathbb{R}} |P(B) - Q(B)|. \quad (4)$$

Combining (3) and (4), we have

$$TV(P, Q) = \frac{1}{2} \int |p - q|.$$

(b) (5 pts.)

Let \mathcal{F} be the σ -field generated by the sets A on the sample space Ω , and

$$\mathcal{C} = T(\mathcal{F}) = \{T(A) : A \in \mathcal{F}\}.$$

Define $T^{-1}(C) = \{\omega \in \Omega : T(\omega) \in C\}$, i.e. the pre-image mapping. By definition,

$$T^{-1}(\mathcal{C}) = \{T^{-1}(C) : C \in \mathcal{C}\} \subseteq \mathcal{F}.$$

Then,

$$\begin{aligned} \sup_{C \in \mathcal{C}} |P(T(X) \in C) - P(T(Y) \in C)| &= \sup_{A \in T^{-1}(\mathcal{C})} |P(X \in A) - P(Y \in A)| \\ &\leq \sup_{A \in \mathcal{F}} |P(X \in A) - P(Y \in A)|. \end{aligned}$$

(c) (5 pts.)

$$\begin{aligned} \int |p \star K - q \star K| &= \int \left| \int p(z)K(x-z)dz - \int q(z)K(x-z)dz \right| dx \\ &= \int \left| \int (p(z) - q(z))K(x-z)dz \right| dx \\ &\leq \int \int |p(z) - q(z)| |K(x-z)| dz dx \\ &\leq \int \int |p(z) - q(z)| |K(x-z)| dx dz && \text{Fubini's theorem} \\ &= \int \left(|p(z) - q(z)| \int |K(x-z)| dx \right) dz \\ &= \int \left(|p(z) - q(z)| \int |K(x)| dx \right) dz && \text{invariant to translation} \\ &= \int |K(x)| dx \int |p(z) - q(z)| dz \\ &= \int |K| \int |p - q| \end{aligned}$$

(d) (10 pts.) Here we can further assume that the density has bounded support, see appendix for a proof without this assumption. By Cauchy inequality,

$$(\int |p - p_n|)^2 \leq \int (p - p_n)^2 \int 1^2 \rightarrow 0,$$

where $\int 1^2$ is finite because density has bounded support.

(e) (5 pts.) We need to show that the integral is finite, $\int \sqrt{p} < +\infty$.

First, the regularly varying tail condition can be translated (not rigorously) as an expression for large value x ,

$$p(tx) = t^r p(x), \forall |x| > B,$$

where $B > 0$ is a constant. Then we decompose the integral into three parts,

$$\int_x \sqrt{p(x)} = \int_{|x| \leq B} \sqrt{p(x)} + \int_{x \geq B} \sqrt{p(x)} + \int_{x \leq B} \sqrt{p(x)},$$

where the first term, integrating on bounded region, is finite. In the following, we argue that the second term $\int_{x \geq B} \sqrt{p(x)}$ is finite, and the third term is also finite using similar argument. By substituting variable $x = Bt$, and using regularly varying tail condition, the second term is,

$$\int_{x \geq B} \sqrt{p(x)} dx = B \int_{t \geq 1} \sqrt{p(tB)} dt = B \int_{t \geq 1} \sqrt{p(B)} t^{r/2} dt.$$

Since $r < -2$, the integral $\int_{t \geq 1} t^{r/2} dt$, is finite.

Appendix

Proof of Claim in Problem 2.

From $\frac{1}{2^p}|a|^p - |b|^p \leq |a - b|^p \leq 2^p|a|^p + 2^p|b|^p$, we have

$$2^{-p}\mathbb{E}[|Z_i|^p] - p_h(x)^p \leq \mathbb{E}[|Z_i - p_h(x)|^p] \leq 2^p\mathbb{E}[|Z_i|^p] + 2^p p_h(x)^p.$$

Then,

$$\begin{aligned} \mathbb{E}[|Z_i|^p] &= \frac{1}{h^p} \int |K|^p \left(\frac{\|x - u\|}{h} \right) p(u) du \\ &= \frac{1}{h^{p-1}} \int |K|^p(\|v\|) p(x + hv) dv. \end{aligned}$$

So as $h \rightarrow 0$, choose any $[a, b]$ such that $|K|^p(\|v\|) > 0$ for some $v \in [a, b]$, then $\int |K|^p(\|v\|) p(x + hv) dv \geq \int_a^b |K|^p(\|v\|) p(x + hv) dv \rightarrow \int_a^b |K|^p(\|v\|) p(x) dv > 0$ by the Bounded Convergence Theorem. Also, $\int |K|^p(\|v\|) p(x + hv) dv \leq \int |K|^p(\|v\|) \sup_x p(x) dv < \infty$, hence $\int |K|^p(\|v\|) p(x + hv) dv = \Theta(1)$, and accordingly,

$$\mathbb{E}[|Z_i|^p] = \Theta\left(\frac{1}{h^{p-1}}\right).$$

Then

$$|p_h(x)| = |\mathbb{E}[Z_i]| \leq \mathbb{E}[|Z_i|] = O(1).$$

Hence

$$\Theta\left(\frac{1}{h^{p-1}}\right) = 2^{-p}\mathbb{E}[|Z_i|^p] - p_h(x)^p \leq \mathbb{E}[|Z_i - p_h(x)|^p] \leq 2^p\mathbb{E}[|Z_i|^p] + 2^p p_h(x)^p = \Theta\left(\frac{1}{h^{p-1}}\right)$$

which implies

$$\mathbb{E}[|Z_i - p_h(x)|^p] = \Theta\left(\frac{1}{h^{p-1}}\right).$$

Proof for Problem 6 (d).

First by $\int (p - p_n)^2 \rightarrow 0$, we claim $p_n \rightarrow p$, a.s.

It's because by contradiction, if there exist set A with $\int 1_A > 0$ such that $p_n(x) \not\rightarrow p(x)$, $\forall x \in A$, then $\int (p - p_n)^2 \geq \int_A (p - p_n)^2 > 0$.

Then note that $\int |p - p_n|$ is bounded,

$$\int 0 \leq \int |p - p_n| \leq \int p + p_n = 2.$$

Thus by Dominated convergence theorem,

$$\int |p - p_n| \rightarrow \int (\lim |p - p_n|) = 0.$$

10/36-702 Statistical Machine Learning Homework #2 Solutions

DUE: 3:00 PM February 22, 2019

Problem 1 [10 pts.]

Consider the data $(X_1, Y_1), \dots, (X_n, Y_n)$ where $X_i \in \mathbb{R}$ and $Y_i \in \mathbb{R}$. Inspired by the fact that $\mathbb{E}[Y|X = x] = \int y p(x, y) dy / p(x)$, define

$$\widehat{m}(x) = \frac{\int y \widehat{p}(x, y) dy}{\widehat{p}(x)}$$

where

$$\widehat{p}(x) = \frac{1}{n} \sum_i \frac{1}{h} K\left(\frac{X_i - x}{h}\right)$$

and

$$\widehat{p}(x, y) = \frac{1}{n} \sum_i \frac{1}{h^2} K\left(\frac{X_i - x}{h}\right) K\left(\frac{Y_i - y}{h}\right).$$

Assume that $\int K(u) du = 1$ and $\int u K(u) du = 0$. Show that $\widehat{m}(x)$ is exactly the kernel regression estimator that we defined in class.

Solution.

$$\begin{aligned} \frac{\int y \cdot \widehat{p}(x, y) dy}{\widehat{p}(x)} &= \frac{\frac{1}{nh^2} \int y \sum K\left(\frac{x-X_i}{h}\right) K\left(\frac{y-Y_i}{h}\right) dy}{\frac{1}{nh} \sum K\left(\frac{x-X_i}{h}\right)} \\ &= \frac{\sum K\left(\frac{x-X_i}{h}\right) \int y \frac{1}{h} K\left(\frac{y-Y_i}{h}\right) dy}{\sum K\left(\frac{x-X_i}{h}\right)} \\ &= \frac{\sum K\left(\frac{x-X_i}{h}\right) Y_i}{\sum K\left(\frac{x-X_i}{h}\right)} \\ &= \widehat{m}(x). \end{aligned}$$

Problem 2 [15 pts.]

Suppose that (X, Y) is bivariate Normal:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \eta \end{pmatrix}, \begin{pmatrix} \sigma^2 & \rho\sigma\tau \\ \rho\sigma\tau & \tau^2 \end{pmatrix}\right).$$

- (a) (5 pts.) Show that $m(x) = \mathbb{E}[Y|X = x] = \alpha + \beta x$ and find explicit expressions for α and β .
- (b) (5 pts.) Find the maximum likelihood estimator $\hat{m}(x) = \hat{\alpha} + \hat{\beta}x$.
- (c) (5 pts.) Show that $|\hat{m}(x) - m(x)|^2 = O_P(n^{-1})$.

Solution.

- (a) Some simple calculations show

$$Y|X = x \sim N\left(\eta + \frac{\tau}{\sigma}\rho(x - \mu), (1 - \rho^2)\tau^2\right),$$

which gives

$$\alpha = \eta - \frac{\tau\rho\mu}{\sigma} \quad \text{and} \quad \beta = \frac{\tau\rho}{\sigma}.$$

- (b) Given a sample $(X_1, Y_1), \dots, (X_n, Y_n)$, the MLEs for the bivariate normal parameters are

$$\begin{aligned} \hat{\mu} &= \bar{X} \\ \hat{\eta} &= \bar{Y} \\ \hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \\ \hat{\tau}^2 &= \frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})^2 \\ \widehat{\text{Cov}}(X, Y) &= \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}). \end{aligned}$$

Note $\beta = \frac{\tau\rho}{\sigma} = \frac{\tau\rho\sigma}{\sigma^2}$. Then by the equivariance property of the MLE,

$$\hat{\beta} = \frac{\widehat{\text{Cov}}(X, Y)}{\hat{\sigma}^2}$$

and

$$\hat{\alpha} = \bar{Y} - \hat{\beta}\bar{X}.$$

Again by equivariance,

$$\hat{m}(x) = \hat{\alpha} + \hat{\beta}x.$$

(c) $\widehat{m}(x)$ is an MLE and satisfies the regularity conditions for asymptotic normality. Therefore,

$$\sqrt{n}(\widehat{m}(x) - m(x)) \rightsquigarrow N(0, I^{-1}(m(x))),$$

which implies

$$\sqrt{n}|\widehat{m}(x) - m(x)| = O_p(1) \implies |\widehat{m}(x) - m(x)|^2 = O_p(n^{-1}).$$

Problem 3 [20 pts.]

Let $m(x) = \mathbb{E}[Y|X = x]$. Let $X \in [0, 1]^d$. Divide $[0, 1]^d$ into cubes B_1, \dots, B_N whose sides have length h . The data are $(X_1, Y_1), \dots, (X_n, Y_n)$. In this problem, treat the X_i 's as fixed. Assume that the number of observations in each bin is positive. Let

$$\widehat{m}(x) = \frac{1}{n(x)} \sum_i Y_i \mathbb{1}(X_i \in B(x))$$

where $B(x)$ is the cube containing x and $n(x) = \sum_i \mathbb{1}(X_i \in B(x))$. Assume that

$$|m(y) - m(x)| \leq L \|x - y\|_2$$

for all x, y . You may further assume that $\sup_x \text{Var}(Y|X = x) < \infty$.

- (a) (10 pts.) Show that

$$|\mathbb{E}[\widehat{m}(x)] - m(x)| \leq C_1 h$$

for some $C_1 > 0$. Also show that

$$\text{Var}(\widehat{m}(x)) \leq \frac{C_2}{n(x)}$$

for some $C_2 > 0$.

- (b) (10 pts.) Now let X be random and assume that X has a uniform density on $[0, 1]^d$. Let $h \equiv h_n = (C \log n/n)^{1/d}$. Show that, for $C > 0$ large enough, $P(\min n_j = 0) \rightarrow 0$ as $n \rightarrow \infty$ where n_j is the number of observations in cube B_j .

Solution.

- (a) We have that X_i are fixed, so that $m(X_i) = Y_i$. Were they not, the below is still applicable by using the law of iterated expectation and the law of total variance.

$$\begin{aligned} |\mathbb{E}[\widehat{m}(x)] - m(x)| &= \left| \mathbb{E}\left[\frac{1}{n(x)} \sum_i Y_i \mathbb{1}_{\{X_i \in B(x)\}} \right] - m(x) \right| \\ &= \left| \frac{1}{n(x)} \sum_i (\mathbb{E}[Y_i] - m(x)) \mathbb{1}_{\{X_i \in B(x)\}} \right| \\ &= \left| \frac{1}{n(x)} \sum_i (m(X_i) - m(x)) \mathbb{1}_{\{X_i \in B(x)\}} \right| \\ &\leq \frac{1}{n(x)} \sum_i |m(X_i) - m(x)| \mathbb{1}_{\{X_i \in B(x)\}} \\ &\leq \frac{1}{n(x)} \sum_i L \sqrt{dh} \cdot \mathbb{1}_{\{X_i \in B(x)\}} \\ &= L \sqrt{dh} \end{aligned}$$

With the first upper bound due to triangular inequality and the second one because, given $x, y \in B_i$:

$$\|x - y\|_2^2 = \sum_{j=1}^d (x_j - y_j)^2 \leq dh^2 \implies \|x - y\|_2 \leq \sqrt{dh}$$

Let $\sup_x \text{Var}(Y|X = x) = M$.

$$\begin{aligned}\text{Var}(\widehat{m}(x)) &= \text{Var}\left(\frac{1}{n(x)} \sum_i Y_i \mathbb{1}_{\{X_i \in B(x)\}}\right) \\ &= \frac{1}{n^2(x)} \sum_i \text{Var}(Y_i) \mathbb{1}_{\{X_i \in B(x)\}} \\ &\leq \frac{M}{n(x)}.\end{aligned}$$

(b)

$$\begin{aligned}P(\min_j n_j = 0) &= P\left(\bigcup_{j=1}^B \{n_j = 0\}\right) \\ &\leq \sum_{j=1}^B P(n_j = 0) \\ &= \sum_{j=1}^B \prod_{i=1}^n \left(1 - P(X_i \in B_j)\right) \\ &= \frac{1}{h^d} (1 - h^d)^n \\ &= \frac{n}{C \log n} \left(1 - \frac{C \log n}{n}\right)^n\end{aligned}$$

Since $B = \frac{1}{h^d}$.¹ Take $C = 1$. Then

$$\begin{aligned}\frac{n}{C \log n} \left(1 - \frac{C \log n}{n}\right)^n &< \frac{n}{C \log n} e^{-\frac{C \log n}{n} \cdot n} \\ &= \frac{n}{C \log n} n^{-C} \\ &= \frac{1}{C \log n} \\ &\rightarrow 0.\end{aligned}$$

¹if we assume $1/h$ is an integer, otherwise we could use that as an upper bound.

Problem 4 [15 pts.]

Consider the RKHS problem

$$\widehat{f} = \operatorname{argmin}_{f \in \mathcal{H}} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2, \quad (1)$$

for some Mercer kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. In this problem, you will prove that the above problem is equivalent to the finite dimensional one

$$\widehat{\alpha} = \operatorname{argmin}_{\alpha \in \mathbb{R}^n} \|y - K\alpha\|_2^2 + \lambda \alpha^T K \alpha, \quad (2)$$

where $K \in \mathbb{R}^{n \times n}$ denotes the kernel matrix $K_{ij} = K(x_i, x_j)$.

Recall that the functions $K(\cdot, x_i)$, $i = 1, \dots, n$ are called the *representers of evaluation*.

Recall that

- $\langle f, K(\cdot, x_i) \rangle_{\mathcal{H}} = f(x_i)$, for any function $f \in \mathcal{H}$
- $\|f\|_{\mathcal{H}}^2 = \sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j)$ for any function $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$.

- (a) (5 pts.) Let $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$, and consider defining a function $\tilde{f} = f + \rho$, where ρ is any function orthogonal to $K(\cdot, x_i)$, $i = 1, \dots, n$. Using the properties of the representers, prove that $\tilde{f}(x_i) = f(x_i)$ for all $i = 1, \dots, n$, and $\|\tilde{f}\|_{\mathcal{H}}^2 \geq \|f\|_{\mathcal{H}}^2$.
- (b) (10 pts.) Conclude from part (a) that in the infinite-dimensional problem (1), we need only consider functions of the form $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$, and that this in turn reduces to (2).

Solution.

- (a) Since $f, \tilde{f} \in \mathcal{H}_K$, for all $i = 1, \dots, n$

$$\begin{aligned} \tilde{f}(x_i) &= \langle \tilde{f}, K(\cdot, x_i) \rangle_{\mathcal{H}_K} \\ &= \langle f, K(\cdot, x_i) \rangle_{\mathcal{H}_K} + \langle \rho, K(\cdot, x_i) \rangle_{\mathcal{H}_K} \\ &= \langle f, K(\cdot, x_i) \rangle_{\mathcal{H}_K} \\ &= f(x_i). \end{aligned}$$

Also, because

$$\begin{aligned} \langle \rho, f \rangle_{\mathcal{H}_K} &= \left\langle \rho, \sum_{i=1}^n \alpha_i K(\cdot, x_i) \right\rangle_{\mathcal{H}_K} \\ &= \sum_{i=1}^n \alpha_i \langle \rho, K(\cdot, x_i) \rangle_{\mathcal{H}_K} \\ &= 0, \end{aligned}$$

we have,

$$\begin{aligned} \|\tilde{f}\|_{\mathcal{H}_K}^2 &= \langle \tilde{f}, \tilde{f} \rangle_{\mathcal{H}_K} \\ &= \langle f, f \rangle_{\mathcal{H}_K} + \langle \rho, \rho \rangle_{\mathcal{H}_K} + 2 \langle \rho, f \rangle_{\mathcal{H}_K} \\ &= \|f\|_{\mathcal{H}_K}^2 + \|\rho\|_{\mathcal{H}_K}^2 \\ &\geq \|f\|_{\mathcal{H}_K}^2. \end{aligned}$$

- (b) For any $\tilde{f} \in \mathcal{H}_K$, let $\tilde{y} = (\tilde{f}(x_1), \dots, \tilde{f}(x_n))^T \in \mathbb{R}^n$. Let $f \in \mathcal{H}_K$ be $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$, where $\alpha = K^{-1}\tilde{y}$. Then

$$\begin{aligned} \langle \tilde{f} - f, K(\cdot, x_i) \rangle_{\mathcal{H}_K} &= \langle \tilde{f}, K(\cdot, x_i) \rangle_{\mathcal{H}_K} - \sum_{j=1}^n \alpha_j \langle K(\cdot, x_j), K(\cdot, x_i) \rangle_{\mathcal{H}_K} \\ &= \tilde{f}(x_i) - \sum_{j=1}^n \alpha_j K(x_i, x_j) \\ &= \tilde{f}(x_i) - [K(K^{-1}\tilde{y})]_i \\ &= \tilde{f}(x_i) - \tilde{f}(x_i) \\ &= 0. \end{aligned}$$

Hence, $\tilde{f} - f \perp K(\cdot, x_i)$ for all $i = 1, \dots, n$, and from (a), this implies $\tilde{f}(x_i) = f(x_i)$ for all $i = 1, \dots, n$, and $\|\tilde{f}\|_{\mathcal{H}_K}^2 \geq \|f\|_{\mathcal{H}_K}^2$, where equality holds if and only if $\tilde{f} = f$. Therefore,

$$\sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}_K}^2 \leq \sum_{i=1}^n (y_i - \tilde{f}(x_i))^2 + \lambda \|\tilde{f}\|_{\mathcal{H}_K}^2,$$

where equality holds if and only if $\tilde{f} = f$. Hence if $\tilde{f} = \operatorname{argmin}_{f \in \mathcal{H}_K} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}_K}^2$, then $\tilde{f} = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$ with $\alpha = K^{-1}\tilde{y}$. So we only need to consider functions of the form $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$. By plugging in, we have

$$\begin{aligned} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}_K}^2 &= \sum_{i=1}^n \left(y_i \sum_{j=1}^n \alpha_j K(x_i, x_j) \right)^2 + \lambda \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(x_i, x_j) \\ &= \|y - K\alpha\|_2^2 + \lambda \alpha^T K \alpha. \end{aligned}$$

Problem 5 [15 pts.]

Let $X = (X(1), \dots, X(d)) \in \mathbb{R}^d$ and $Y \in \mathbb{R}$. In the questions below, make any reasonable assumptions that you need but state your assumptions.

- (a) (5 pts.) Prove that $\mathbb{E}(Y - m(X))^2$ is minimized by choosing $m(x) = \mathbb{E}(Y|X = x)$.
- (b) (5 pts.) Find the function $m(x)$ that minimizes $\mathbb{E}|Y - m(X)|$. (You can assume that the conditional cdf $F(y|X = x)$ is continuous and strictly increasing, for every x .)
- (c) (5 pts.) Prove that $\mathbb{E}(Y - \beta^T X)^2$ is minimized by choosing $\beta_* = B^{-1}\alpha$ where $B = \mathbb{E}(XX^T)$ and $\alpha = (\alpha_1, \dots, \alpha_d)$ and $\alpha_j = \mathbb{E}(YX(j))$.

Solution.

- (a) Let $g(x)$ be any function of x . Then

$$\begin{aligned} \mathbb{E}(Y - g(X))^2 &= \mathbb{E}(Y - m(X) + m(X) - g(X))^2 \\ &= \mathbb{E}(Y - m(X))^2 + \mathbb{E}(m(X) - g(X))^2 + 2\mathbb{E}((Y - m(X))(m(X) - g(X))) \\ &\geq \mathbb{E}(Y - m(X))^2 + 2\mathbb{E}((Y - m(X))(m(X) - g(X))) \\ &= \mathbb{E}(Y - m(X))^2 + 2\mathbb{E}\left((Y - m(X))(m(X) - g(X)) \middle| X\right) \\ &= \mathbb{E}(Y - m(X))^2 + 2\mathbb{E}\left((\mathbb{E}(Y|X) - m(X))(m(X) - g(X))\right) \\ &= \mathbb{E}(Y - m(X))^2 + 2\mathbb{E}\left((m(X) - \mathbb{E}(Y|X))(m(X) - g(X))\right) \\ &= \mathbb{E}(Y - m(X))^2 \end{aligned}$$

- (b) Let $g(x)$ be any function of x . Recall that

$$\mathbb{E}[|Y - g(X)|] = \mathbb{E}\{\mathbb{E}[|Y - g(X)|] | X\}.$$

The idea is to choose c such that $\mathbb{E}[|Y - c| | X = x]$ is minimized. Now define:

$$r(c) = \mathbb{E}[|Y - c| | X = x] = \int |y - c| p_{Y|X=x}(y) dy.$$

The function $h_y(c) = |y - c|$ is differentiable everywhere except when $y = c$. Thus for $c \neq y$

$$h'_y(c) = \begin{cases} 1 & c > y \\ -1 & c < y \end{cases} = \mathbb{1}(c > y) - \mathbb{1}(c < y).$$

Since Y is continuous and has a density function, $P(Y = c) = 0$. So to minimize $r(c)$ we can differentiate under the integral sign and set the derivative equal to 0 to obtain:

$$\begin{aligned} r'(c) &= \int h'_y(c) p_{Y|X=x}(y) dy = \int_{-\infty}^c p_{Y|X=x}(y) dy - \int_c^{\infty} p_{Y|X=x}(y) dy \\ &= 2 \int_{-\infty}^c p_{Y|X=x}(y) dy - 1 = 0 \\ \iff & \int_{-\infty}^c p_{Y|X=x}(y) dy = \frac{1}{2}, \end{aligned}$$

so that $c = m(x)$, which is the median of $p_{Y|X=x}(y)$. It is a minimum since $r'(c) < 0$ for $c < m(x)$ and $r'(c) > 0$ for $c > m(x)$. Since m minimizes $\mathbb{E}[|Y - c| \mid X = x]$ at every x for any g we get

$$\mathbb{E}[|Y - g(X)| - |Y - m(X)| \mid X = x] \geq 0$$

which implies

$$R(g) - R(m) = \mathbb{E}[|Y - g(X)| - |Y - m(X)|] = \mathbb{E}\{\mathbb{E}[|Y - g(X)| - |Y - m(X)|] \mid X]\} \geq 0.$$

(c) By setting the first derivative of the loss function equal to 0 we obtain:

$$\begin{aligned} \frac{\partial R(\beta)}{\partial \beta} &= 0 \\ \implies \frac{\partial \mathbb{E}(Y - \beta^T X)^2}{\partial \beta} &= 0 \\ \implies \mathbb{E}[-2X(Y - \beta^T X)] &= 0 \\ \implies 2B\beta - 2\alpha &= 0 \\ \implies \beta_* &= B^{-1}\alpha, \end{aligned}$$

where we can exchange the derivative and expectation by the dominated convergence theorem. The loss function $R(\beta)$ is strictly convex so β_* is its unique minimum.

Problem 6 [25 pts.]

Consider the many Normal means problem where we observe $Y_i \sim N(\theta_i, 1)$ for $i = 1, \dots, d$. Let $\hat{\theta}$ minimize the penalized loss

$$\sum_i (Y_i - \theta_i)^2 + \lambda J(\theta).$$

Find an explicit form for $\hat{\theta}$ in three cases: (i) (10 pts.) $J(\beta) = \|\theta\|_0$, (ii) (10 pts.) $J(\beta) = \|\theta\|_1$ and (iii) (5 pts.) $J(\beta) = \|\theta\|_2^2$.

Solution.

(i) Note that

$$\sum_i (Y_i - \theta_i)^2 + \lambda \|\theta\|_0 = \sum_{j=1}^d \left((Y_j - \theta_j)^2 + \lambda \mathbb{1}(\theta_j \neq 0) \right).$$

Then for each term i ,

$$\begin{aligned} (Y_i - \theta_i)^2 + \lambda \mathbb{1}(\theta_i \neq 0) &\geq Y_i^2 \mathbb{1}(\theta_i = 0) + \lambda \mathbb{1}(\theta_i \neq 0) \\ &\geq \min \left\{ Y_i^2, \lambda \right\} \end{aligned}$$

and equality holds if and only if

$$\hat{\theta}_i = \begin{cases} 0 & \text{if } Y_i^2 < \lambda \\ 0 \text{ or } Y_i & \text{if } Y_i^2 = \lambda \\ Y_i & \text{if } Y_i^2 > \lambda. \end{cases}$$

Hence

$$\begin{aligned} \sum_i (Y_i - \theta_i)^2 + \lambda \|\theta\|_0 &= \sum_{j=1}^d \left((Y_j - \theta_j)^2 + \lambda \mathbb{1}(\theta_j \neq 0) \right) \\ &\geq \sum_{j=1}^d \min \left\{ Y_j^2, \lambda \right\} \end{aligned}$$

and equality holds if and only if

$$\hat{\theta}_i = \begin{cases} 0 & \text{if } |Y_i| < \sqrt{\lambda} \\ 0 \text{ or } Y_i & \text{if } |Y_i| = \sqrt{\lambda} \\ Y_i & \text{if } |Y_i| > \sqrt{\lambda}. \end{cases}$$

(ii) First write

$$\min_{\theta} \sum_i (Y_i - \theta_i)^2 + \lambda \|\theta\|_1 = \min_{\theta} \sum_i (-2Y_i \theta_i + \theta_i^2 + \lambda |\theta_i|).$$

Now note it is simply equivalent to

$$\begin{aligned} & \min_{\theta_i} -2Y_i \theta_i + \theta_i^2 + \lambda |\theta_i| \\ \iff & \min_{\theta_i} -2\widehat{\theta}_i^{OLS} \theta_i + \theta_i^2 + \lambda |\theta_i| \end{aligned}$$

for all $i = 1, \dots, d$.

When $\widehat{\theta}_i^{OLS} \geq 0$, then $\widehat{\theta}_i \geq 0$ so

$$-2\widehat{\theta}_i^{OLS} \theta_i + \theta_i^2 + \lambda |\theta_i| = -2\widehat{\theta}_i^{OLS} \theta_i + \theta_i^2 + \lambda \theta_i.$$

Differentiating with respect to θ_i , setting equal to zero, and solving gives

$$\widehat{\theta}_i = \left(\widehat{\theta}_i^{OLS} - \frac{\lambda}{2} \right) \mathbb{1}_{\{\widehat{\theta}_i^{OLS} \geq \frac{\lambda}{2}\}}.$$

When $\widehat{\theta}_i^{OLS} \leq 0$, the analogous steps give

$$\widehat{\theta}_i = \left(\widehat{\theta}_i^{OLS} + \frac{\lambda}{2} \right) \mathbb{1}_{\{\widehat{\theta}_i^{OLS} \leq -\frac{\lambda}{2}\}},$$

Putting them together gives

$$\widehat{\theta}_i = \begin{cases} \widehat{\theta}_i^{OLS} - \frac{\lambda}{2} & \widehat{\theta}_i^{OLS} \geq \frac{\lambda}{2} \\ 0 & \widehat{\theta}_i^{OLS} \in \left(-\frac{\lambda}{2}, \frac{\lambda}{2} \right) \\ \widehat{\theta}_i^{OLS} + \frac{\lambda}{2} & \widehat{\theta}_i^{OLS} \leq -\frac{\lambda}{2} \end{cases}$$

(iii) Here the objective function is differentiable everywhere. Taking the gradient w.r.t. θ we have

$$\nabla_{\theta} \left(\sum_i (Y_i - \theta_i)^2 + \lambda \|\theta\|_2^2 \right) = \sum_i (-2Y_i \theta_i + 2\lambda \theta_i).$$

Setting this equal to 0 and solving for θ gives

$$\widehat{\theta}_i = \frac{Y_i}{1 + \lambda}. \tag{3}$$

Since the objective is strictly convex, (3) is the unique solution.

36-708 Statistical Machine Learning Homework #3 Solutions

DUE: March 29, 2019

Problem 1 [9 pts.]

Get the iris data. (In R, use `data(iris)`.) There are 150 observations. The outcome is “Species” which has three values. The goal is to predict Species using the four covariates. Compare the following classifiers: (i) LDA [3 pts.], (ii) logistic regression [3 pts.], (iii) nearest neighbors [3 pts.]. Note that you will need to figure out a way to deal with three classes when using logistic regression. Explain how you handled this. Summarize your results.

Solution.

In order to deal with multiple classes for logistic regression one can either fit a logistic regression for each of the classes versus the others or use a multinomial logistic regression model. We will use the latter. In these settings the output is a 3 dimensional vector (one per classes) (π_1, π_2, π_3) , such that $\sum_i \pi_i = 1$. The model learns β_1 and β_2 such that:

$$\log\left(\frac{\pi_1}{1 - \pi_1 - \pi_2}\right) = X_i^T \beta_1 \quad \log\left(\frac{\pi_2}{1 - \pi_1 - \pi_2}\right) = X_i^T \beta_2$$

We will use the R package `nnet` which uses neural networks.

```
set.seed(7)
data(iris)

test_size <- 0.4
test_idx <- sample(nrow(iris), size = as.integer(nrow(iris) * test_size))

X_train <- iris[-test_idx, 1:4]
y_train <- iris[-test_idx, 5]
X_test <- iris[test_idx, 1:4]
y_test <- iris[test_idx, 5]

##### LDA
require(MASS)
fit_lda <- lda(X_train, y_train)
pred_lda <- predict(fit_lda, X_test)
lda_acc <- sum(pred_lda$class == y_test)/(length(y_test))
table(pred_lda$class, y_test)

##### Multinomial Regression
require(nnet)
fit_logreg <- multinom(Species~Sepal.Length +Sepal.Width +Petal.Length +Petal.Width,
                         data=iris[-test_idx,])
pred_logreg <- predict(fit_logreg, X_test)
mult_logreg_acc <- sum(pred_logreg == y_test)/(length(y_test))
table(pred_logreg, y_test)
```

```
##### KNN
require(class)
pred_knn <- knn(X_train, X_test, cl=y_train, k=3)
knn_acc <- sum(pred_knn == y_test)/(length(y_test))
table(pred_knn, y_test)
```

LDA and K-nearest neighbors classification achieves 96.67% accuracy, while multinomial logistic regression achieves 95% accuracy. Given the low number of samples, the three models can be considered equivalent in terms of performance over this dataset. A further analysis changing the random split between training and testing set confirms this hypothesis.

Problem 2 [8 pts.]

Use the iris data again but throw away the Species variable. Use k -means⁺⁺ clustering [4 pts.] and mean-shift clustering [4 pts.]. Compare the clusterings to the true group defined by Species. Which method worked better?

Solution.

We use the R packages `LICORS` and `LPCM` to run k -means⁺⁺ and mean-shift clustering.

```
data(iris)
X <- iris[, 1:4]
y <- iris[, 5]

##### K-MEANS++
library(LICORS)
set.seed(7)
clustering_kmpp <- kmeanspp(X, k = 3, iter.max = 300, nstart = 10)

# Getting the labels of the clustering
table_results_kmpp <- table(clustering_kmpp$cluster, y)
label_kmpp <- apply(table_results_kmpp, 2, which.max)
clustering_labels <- factor(clustering_kmpp$cluster,
                             labels = names(label_kmpp)[order(label_kmpp)]))

# Calculating accuracy
acc_kmpp <- sum(y == clustering_labels)/nrow(X)
print(acc_kmpp)

##### Mean Shift
library(LPCM)
set.seed(7)
clustering_ms <- ms(X, h=0.11)

# Getting the labels of the clustering
table_results_ms <- table(clustering_ms$cluster.label, y)
label_ms <- apply(table_results_ms, 2, which.max)
clustering_labels <- factor(clustering_ms$cluster.label,
                            labels = names(label_ms)[order(label_ms)]))

# Calculating accuracy
acc_ms <- sum(y == clustering_labels)/nrow(X)
print(acc_ms)
```

As in the mean-shift clustering algorithm we cannot input the number of clusters, the algorithm struggles finding exactly only three clusters. We have used the value $h = 0.11$ for the bandwidth, found via using a validation set, to get exactly three clusters and be able to compare the two methods apple-to-apple. K-means seems to be performing better with an accuracy of 89% against an accuracy of 68%.

Problem 3 [9 pts.]

Download the data from <http://www-bcf.usc.edu/~gareth/ISL/Ch10Ex11.csv>. This is a gene expression dataset. There are 40 tissue samples with measurements on 1,000 genes. The first 20 data points are from healthy people. The second 20 data points are from diseased people.

- (a) [3 pts.] Use sparse logistic regression to classify the subject. (You may use the function `glmnet` in R if you like.) Explain how you chose λ . Summarize your findings;
- (b) [3 pts.] Now use a Sparse Additive Model as described in class. Summarize your findings;
- (c) [3 pts.] Now suppose we don't know which are healthy and which are diseased. Apply clustering to put the data into two groups. Applying k -means clustering may not work well because the dimension is so high. Instead, you will need to do some sort of dimension reduction or sparse clustering. One very simple method is Sparse Alternate Similarity (arXiv:1602.07277). But you may use any method you like. Describe what you chose to do and what the results are.

Solution.

For sparse logistic regression, we use the R package `glmnet`, and we choose the best λ via cross validation. For sparse additive models we use the R package `SAM`, in which we run for a series of λ and we select the minimum. For the clustering step we use sparse clustering (Witten, D.M. and Tibshirani, R. *A framework for feature selection in clustering*, 2010) with the R package `sparcl`, in which the best bound for the L_1 norm is selected via a permutation approach and the number of clusters is set to 2. All methods perfectly separate the training data. As a note, one could have further split the data into training and testing but, given the low sample size, the most sensible approach would be to use the leave-one-out mis-classification rate as performance metric.

```
# Read in data
X <- read.csv("http://www-bcf.usc.edu/~gareth/ISL/Ch10Ex11.csv", header=F)

# Generate classes
y <- rep(0,40)
y[21:40] <- 1

##### Sparse Logistic Regression
require(glmnet)

# Tuning Lambda via CV
cvfit <- cv.glmnet(t(X), y, alpha=1, family="binomial", nfold=10)

# Fitting Sparse Logistic Regression
sparse_log_reg <- glmnet(t(X), y, family = "binomial", alpha = 1,
                          lambda = cvfit$lambda.min)
coeff_sparse_log_reg <- as.matrix(coef(sparse_log_reg))
print(nrow(X) - sum(coeff_sparse_log_reg==0)) #Number of >0 coefficients

proba_sparse_log_reg <- predict(object = sparse_log_reg, newx =t(X), type = "response")
pred_sparse_log_reg <- ifelse(proba_sparse_log_reg>0.5, 1, 0)

acc_sparse_log_reg <- sum(y == pred_sparse_log_reg)/length(y)
```

```
print(acc_sparse_log_reg)

##### Sparse Additive Models
require(SAM)

# Fit SAM for 100 lambda - pick the best
sam_model <- samLL(X = as.matrix(t(X)),
                     y = as.matrix(y),
                     p = 3, nlambda=100)

sam_final <- samLL(X = as.matrix(t(X)),
                     y = as.matrix(y),
                     lambda = min(sam_model$lambda), p = 3)

# Prediction
pred_sam <- as.data.frame(predict(object = sam_final,
                                      newdata = t(X)))

acc_sam <- sum(y == pred_sam)/length(y)
print(acc_sam)

##### Sparse Clustering
library(sparcl)

best_bound_ift <- KMeansSparseCluster.permute(t(x), K=2, wbound=seq(1.1, 100,
                                                               length.out=20))
sparse_cluster <- KMeansSparseCluster(x = t(X), K = 2, wbounds = best_bound_ift$bestw)
acc_sparcl <- sum(y == (as.numeric(sparse_cluster[[1]]$Cs) - 1))/length(y)
print(acc_sparcl)
```

Problem 4 [10 pts.]

Let $X = (X_1, \dots, X_d) \in \mathbb{R}^d$. Suppose that $X \sim N(\mu, \Sigma)$. Let $\Omega = \Sigma^{-1}$. Let $j \neq k$ be integers such that $1 \leq j < k \leq d$. Let $Z = (X_s : s \neq j, k)$.

- (a) [3 pts.] Show that the distribution of $(X_j, X_k)|Z$ is $N(a, B)$ and find a and B explicitly.
- (b) [4 pts.] Show that $X_j \perp\!\!\!\perp X_k|Z$ if and only if $\Omega_{jk} = 0$.
- (c) [3 pts.] Now let $X_1, \dots, X_n \sim N(\mu, \Sigma)$. Find the mle $\hat{\Omega}$.

Solution.

(a) More generally, by Theorem 1, for any random vectors $\mathbf{X}_1 \subset X$ and $\mathbf{X}_2 = X \setminus \mathbf{X}_1$, $\mathbf{X}_2|\mathbf{X}_1$ follows a multivariate normal distribution.

Specifically, if $\mathbf{X}_1 \in \mathbb{R}^r$ and $\mathbf{X}_2 \in \mathbb{R}^s = \mathbb{R}^{d-r}$, then

$$\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1 \sim N(\boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}),$$

where (if necessary) we have reordered X so that

$$X = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} \quad \text{and} \quad \mu = \mathbb{E}(X) = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \quad \text{and} \quad \Sigma = \text{Cov}(X) = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}.$$

Proof.

$$i. \quad \mathbb{E}(\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1) = \boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1)$$

Let

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} & 0 \\ -\Sigma_{21}\Sigma_{11}^{-1} & \mathbf{I} \end{pmatrix}.$$

\mathbf{A} is full-rank so, by Theorem 2,

$$AX = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix}$$

follows a multivariate normal distribution, where

$$\tilde{\mathbf{X}}_2 = \mathbf{X}_2 - \Sigma_{21}\Sigma_{11}^{-1}\mathbf{X}_1.$$

Now,

$$\begin{aligned} \text{Cov}(\mathbf{X}_1, \tilde{\mathbf{X}}_2) &= \text{Cov}(\mathbf{X}_1, \mathbf{X}_2 - \Sigma_{21}\Sigma_{11}^{-1}\mathbf{X}_1) \\ &= \text{Cov}(\mathbf{X}_1, \mathbf{X}_2) - \Sigma_{21}\Sigma_{11}^{-1}\text{Cov}(\mathbf{X}_1, \mathbf{X}_1) \\ &= \Sigma_{12} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{11} \\ &= \Sigma_{12} - \Sigma_{12} \\ &= 0, \end{aligned}$$

(a) so \mathbf{X}_1 and $\tilde{\mathbf{X}}_2$ are uncorrelated, and thus independent by Theorem 4. Hence,

$$\begin{aligned}
 \mathbb{E}(\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1) &= \mathbb{E}(\tilde{\mathbf{X}}_2 + \Sigma_{21}\Sigma_{11}^{-1}\mathbf{X}_1|\mathbf{X}_1 = \mathbf{x}_1) \\
 &= \mathbb{E}(\tilde{\mathbf{X}}_2|\mathbf{X}_1 = \mathbf{x}_1) + \Sigma_{21}\Sigma_{11}^{-1}\mathbf{x}_1 \\
 &= \mathbb{E}(\tilde{\mathbf{X}}_2) + \Sigma_{21}\Sigma_{11}^{-1}\mathbf{x}_1 \\
 &= \mathbb{E}(\mathbf{X}_2 - \Sigma_{21}\Sigma_{11}^{-1}\mathbf{X}_1) + \Sigma_{21}\Sigma_{11}^{-1}\mathbf{x}_1 \\
 &= \boldsymbol{\mu}_2 - \Sigma_{21}\Sigma_{11}^{-1}\boldsymbol{\mu}_1 + \Sigma_{21}\Sigma_{11}^{-1}\mathbf{x}_1 \\
 &= \boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1). \quad \checkmark
 \end{aligned}$$

ii. $\text{Cov}(\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1) = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$

Again using the fact that \mathbf{X}_1 and $\tilde{\mathbf{X}}_2$ are independent, we have

$$\begin{aligned}
 &\text{Cov}(\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1) \\
 &= \text{Cov}(\tilde{\mathbf{X}}_2 + \underbrace{\Sigma_{21}\Sigma_{11}^{-1}\mathbf{X}_1}_{\text{fixed}}|\mathbf{X}_1 = \mathbf{x}_1) \\
 &= \text{Cov}(\tilde{\mathbf{X}}_2|\mathbf{X}_1 = \mathbf{x}_1) \\
 &= \text{Cov}(\tilde{\mathbf{X}}_2) \\
 &= \text{Cov}(\mathbf{X}_2 - \Sigma_{21}\Sigma_{11}^{-1}\mathbf{X}_1) \\
 &= \text{Cov}(\mathbf{X}_2, \mathbf{X}_2) - \Sigma_{21}\Sigma_{11}^{-1}\text{Cov}(\mathbf{X}_1, \mathbf{X}_2) - \text{Cov}(\mathbf{X}_2, \mathbf{X}_1)(\Sigma_{21}\Sigma_{11}^{-1})^T + \Sigma_{21}\Sigma_{11}^{-1}\text{Cov}(\mathbf{X}_1, \mathbf{X}_1)(\Sigma_{21}\Sigma_{11}^{-1})^T \\
 &= \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} - \Sigma_{21}(\Sigma_{11}^{-1})^T\Sigma_{21}^T + \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{11}(\Sigma_{11}^{-1})^T\Sigma_{21}^T \\
 &= \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} + \Sigma_{21}(\Sigma_{11}^T)^{-1}\Sigma_{21}^T \\
 &= \Sigma_{22} - 2\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} + \Sigma_{21}(\Sigma_{11})^{-1}\Sigma_{21} \\
 &= \Sigma_{22} - 2\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{21} + \Sigma_{21}(\Sigma_{11})^{-1}\Sigma_{21} \\
 &= \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{21}. \quad \checkmark
 \end{aligned}$$

Hence,

$$\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1 \sim N(\boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}).$$

Notice that the distribution of $(X_j, X_k)|Z$ is given by the special case where

$$\mathbf{X}_1 = (X_s : s \neq j, k) \quad \text{and} \quad \mathbf{X}_2 = (X_j, X_k). \quad \blacksquare$$

(b) As in part (a), let us first reorder X so that

$$X = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} \quad \text{and} \quad \Sigma = \text{Cov}(X) = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix},$$

where

$$\mathbf{X}_1 = (X_s : s \neq j, k) \quad \text{and} \quad \mathbf{X}_2 = (X_j, X_k).$$

Similarly, we can partition the (unknown) matrix $\Omega \in \mathbb{R}^{d \times d}$ as

$$\Omega = \left(\begin{array}{c|c} \Omega_{11} & \Omega_{12} \\ \hline \Omega_{21} & \Omega_{22} \end{array} \right),$$

so that

$$\Omega_{22} = \begin{pmatrix} \Omega_{jj} & \Omega_{jk} \\ \Omega_{kj} & \Omega_{kk} \end{pmatrix}.$$

By definition, and using the fact that Σ is symmetric (and thus, so is Ω), we have

$$\Sigma\Omega = \left(\begin{array}{c|c} \Sigma_{11}\Omega_{11} + \Sigma_{12}\Omega_{12}^T & \Sigma_{11}\Omega_{12} + \Sigma_{12}\Omega_{22} \\ \hline \Sigma_{12}^T\Omega_{11} + \Sigma_{22}\Omega_{12}^T & \Sigma_{12}^T\Omega_{12} + \Sigma_{22}\Omega_{22} \end{array} \right) = \left(\begin{array}{c|c} \mathbf{I}_{d-2} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_2 \end{array} \right).$$

Setting each block equal,

$$\Sigma_{11}\Omega_{11} + \Sigma_{12}\Omega_{12}^T = \mathbf{I}_{d-2} \implies \Omega_{11} = \Sigma_{11}^{-1} - \Sigma_{11}^{-1}\Sigma_{12}\Omega_{12}^T \quad (1)$$

$$\Sigma_{12}^T\Omega_{11} + \Sigma_{22}\Omega_{12}^T = \mathbf{0} \implies \Omega_{12}^T = -\Sigma_{22}^{-1}\Sigma_{12}^T\Omega_{11}$$

$$\Sigma_{11}\Omega_{12} + \Sigma_{12}\Omega_{22} = \mathbf{0} \implies \Omega_{12} = -\Sigma_{11}^{-1}\Sigma_{12}\Omega_{22} \quad (1)$$

$$\Sigma_{12}^T\Omega_{12} + \Sigma_{22}\Omega_{22} = \mathbf{I}_2 \implies \Omega_{22} = \Sigma_{22}^{-1} - \Sigma_{22}^{-1}\Sigma_{12}^T\Omega_{12}. \quad (2)$$

Plugging (1) into (2), we get

$$\begin{aligned} \Omega_{22} &= \Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{12}^T\Sigma_{11}^{-1}\Sigma_{12}\Omega_{22} \\ &\implies \mathbf{I}_2 = \Sigma_{22}^{-1}\Omega_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{12}^T\Sigma_{11}^{-1}\Sigma_{12} \\ &\implies (\mathbf{I}_2 - \Sigma_{22}^{-1}\Sigma_{12}^T\Sigma_{11}^{-1}\Sigma_{12})\Omega_{22} = \Sigma_{22}^{-1} \\ &\implies (\Sigma_{22} - \Sigma_{12}^T\Sigma_{11}^{-1}\Sigma_{12})\Omega_{22} = \mathbf{I}_2 \\ &\implies \Omega_{22} = (\Sigma_{22} - \Sigma_{12}^T\Sigma_{11}^{-1}\Sigma_{12})^{-1} \\ &\implies \Omega_{22} = (\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{21}^T)^{-1}. \end{aligned}$$

Using part (a) we see

$$\Omega_{22}^{-1} = \begin{pmatrix} \Omega_{jj} & \Omega_{jk} \\ \Omega_{kj} & \Omega_{kk} \end{pmatrix}^{-1} = \text{Cov}(\mathbf{X}_2 | \mathbf{X}_1 = \mathbf{x}_1).$$

“ \implies ” Suppose $X_j \perp\!\!\!\perp X_k | Z$. Then X_j and X_k are uncorrelated given Z . That is, the off-diagonal elements of the 2×2 matrix $\text{Cov}(\mathbf{X}_2 | \mathbf{X}_1 = \mathbf{x}_1)$ are zero. And the inverse of any diagonal matrix is diagonal so $\Omega_{jk} = \Omega_{kj} = 0$.

“ \Leftarrow ” Suppose $\Omega_{jk} = 0$. $\Omega^T = (\Sigma^{-1})^T = (\Sigma^T)^{-1} = \Sigma^{-1} = \Omega$, so $\Omega_{kj} = 0$ as well. That is, Ω_{22} is diagonal. Therefore, its inverse $\text{Cov}(\mathbf{X}_2 | \mathbf{X}_1 = \mathbf{x}_1)$ is also diagonal, which implies X_j and X_k are uncorrelated given Z . By Theorem 4, X_j and X_k are independent given Z . ■

(c) The log likelihood in terms of Ω is,

$$l \propto \frac{n}{2} \log |\Omega| - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^T \Omega (x_i - \mu).$$

MLE can be get by taking derivative with respect to Ω and set it to zero,

$$\frac{n}{2} \Omega^{-1} - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T = 0,$$

that is,

$$\hat{\Omega} = \left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T \right)^{-1}.$$

Alternatively, the MLE for Ω , the inverse of Σ , is the inverse of MLE for Σ ,

$$\hat{\Omega} = \hat{\Sigma}^{-1} = (\hat{\Sigma})^{-1} = \left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T \right)^{-1}.$$

Problem 5 [12 pts.]

Let $X = (X_1, X_2, X_3, X_4, X_5)$ be a random vector distributed as $X \sim N(0, \Sigma)$ where

$$\Sigma^{-1} = \begin{pmatrix} 3 & 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 1 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{pmatrix}.$$

- (a) [1 pts.] What is the graph for X , viewed as an undirected graphical model?
- (b) [2 pts.] List the maximal cliques of the graph.
- (c) [4 pts.] Which of the following independence statements are true?
- (a) $X_2 \perp\!\!\!\perp X_3 | X_1, X_2$
 - (b) $X_3 \perp\!\!\!\perp X_4 | X_5$
 - (c) $\{X_1, X_2\} \perp\!\!\!\perp X_3 | X_4, X_5$
 - (d) $X_1 \perp\!\!\!\perp X_5 | X_3$
- (d) [2 pts.] List the local Markov properties for this graphical model.
- (e) [3 pts.] Simulate 100 observations from this model. Construct a graph using hypothesis testing. Report your results. Include your code.

Solution.

- (a) The edges can be seen directly from Σ^{-1} . That is,

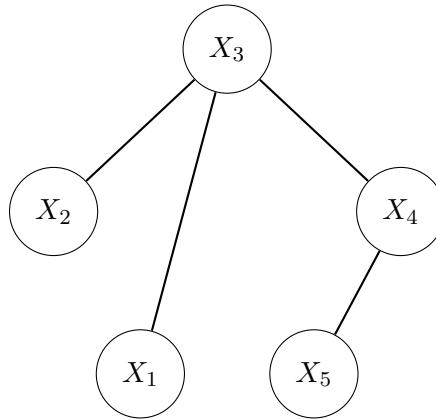


Figure 1. Conditional independence graph of $X = (X_1, X_2, X_3, X_4, X_5)$

- (b) By definition, the maximal cliques are $\{1, 3\}, \{2, 3\}, \{3, 4\}, \{4, 5\}$.
- (c) (Statement (a) has an typo and is ignored)

These statements can be verified or falsified by simply considering the graph in part (a). To see the conditional independence between set of nodes A and B conditioning on C , check if there is a connecting path between A and B when the nodes in C are blocked. As a result,

- (b) FALSE;
- (c) FALSE;
- (d) TRUE.

- (d) Local Markov property is :

$$\forall s \in V, p(x_s|x_t, t \neq s) = p(x_s|x_t, t \in N(s)).$$

Hence, in our case, Local Markov properties is as below :

$$\begin{aligned} X_1|X_2, X_3, X_4, X_5 &\stackrel{d}{=} X_1|X_3 \\ X_2|X_1, X_3, X_4, X_5 &\stackrel{d}{=} X_2|X_3 \\ X_3|X_1, X_2, X_4, X_5 &\stackrel{d}{=} X_3|X_1, X_2, X_4 \\ X_4|X_1, X_2, X_3, X_5 &\stackrel{d}{=} X_4|X_3, X_5 \\ X_5|X_1, X_2, X_3, X_4 &\stackrel{d}{=} X_5|X_4. \end{aligned}$$

-
- (e) As provided in the lecture notes, you can either construct a marginal correlation graph or a partial correlation graph. We present the code for a partial correlation graph following the normal approximation testing described in page 11. The produced graph is consistent with the graph in part (a).

```
library(MASS)
#generate sample
d = 5; n = 100; alpha = 0.05; m = d*(d-1)/2
omega <- matrix(c(3,0,1,0,0,
                  0,3,1,0,0,
                  1,1,3,1,0,
                  0,0,1,3,1,
                  0,0,0,1,3), ncol = 5, byrow = TRUE)
X <- mvrnorm(n, mu = rep(0,d), Sigma = solve(omega))

#estimate matrix R
S_n <- 1/n * t(X)%*%X
hatOmega <- solve(S_n)
Rmat <- -hatOmega/sqrt(outer(diag(hatOmega), diag(hatOmega)))

#test edge
Z <- 1/2*log((1+Rmat)/(1-Rmat))
edge <- abs(Z) > (qnorm(1 - alpha/(2*m))/sqrt(n - d - 1))
```

Problem 6 [8 pts.]

Let $X = (X_1, \dots, X_4)$ where each variable is binary. Suppose the probability function is

$$\log p(x) = \psi_{\emptyset} + \psi_1(x_1) + \psi_{12}(x_1, x_2) + \psi_{13}(x_1, x_3) + \psi_{24}(x_2, x_4) + \psi_{34}(x_3, x_4).$$

- (a) [3 pts.] Draw the implied graph;
- (b) [3 pts.] Write down all the independence and conditional independence relations implied by the graph;
- (c) [2 pts.] Is the model graphical? Is the model hierarchical?

Solution.

- (a) The implied graph is

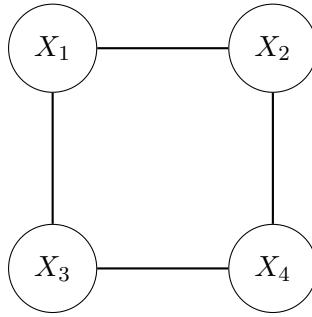


Figure 2. Implied graph of $X = (X_1, X_2, X_3, X_4)$

- (b) From Theorem 9 in lectures notes, we have $X_1 \perp\!\!\!\perp X_4 | X_2, X_3$ and $X_2 \perp\!\!\!\perp X_3 | X_1, X_4$.
- (c) Not graphical, not hierarchical. This model satisfies $\psi_1(x_1) = 0$ but $\psi_{12}(x_1, x_2) \neq 0$, so the model is not hierarchical. And hence this model is not graphical as well, by Lemma 10 in the Graphical Models lecture notes.

Problem 7 [8 pts.]

Let X_1, \dots, X_4 be binary. Draw the independence graphs corresponding to the following log-linear models (where $\alpha \in \mathbb{R}$). Also, identify whether each is graphical and/or hierarchical (or neither).

- (a) [2 pts.] $\log p(x) = \alpha + 11x_1 + 2x_2 + 3x_3$
- (b) [2 pts.] $\log p(x) = \alpha + 2x_2 + 1.5x_3 + 17x_4 + 12x_2x_3 + 78x_2x_4 + 3x_3x_4 + 32x_2x_3x_4$
- (c) [2 pts.] $\log p(x) = \alpha + 9x_1 + 2x_2 + 1.5x_3 + 17x_4 + 12x_2x_3 + 3x_3x_4 + x_1x_4 + 2x_1x_2$
- (d) [2 pts.] $\log p(x) = \alpha + 115x_1x_2x_3x_4.$

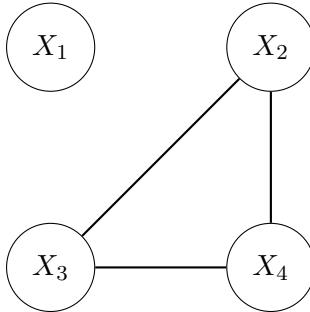
Solution.

- (a) Hierarchical, but not graphical.

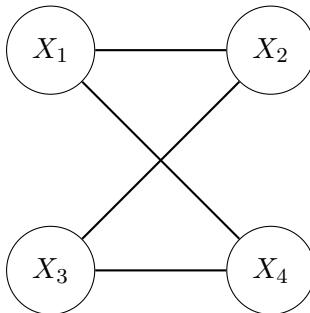


Note: X_4 is a clique, but $\beta_4 = 0$.

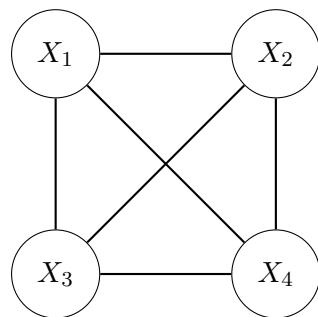
- (b) Hierarchical, but not graphical.



- (c) Graphical and hierarchical.



(d) Not graphical, nor hierarchical.



Problem 8 [10 pts.]

Consider the log-linear model

$$\log p(x) = \beta_0 + x_1 x_2 + x_2 x_3 + x_3 x_4.$$

Simulate $n = 1000$ random vectors from this distribution. (Show your code.) Fit the model

$$\log p(x) = \beta_0 + \sum_j \beta_j x_j + \sum_{k < \ell} \beta_{k\ell} x_k x_\ell$$

using maximum likelihood. Report your estimators. Use hypothesis testing to decide which parameters are non-zero. Compare the selected model to the true model.

Solution.

To simulate random vectors from the log-linear distribution we simply convert to the corresponding multinomial and sample from it. We then fit the requested model using `glm` command with a log-link function (`family = "poisson"`). The results for $n = 1000$ are given in Table I.

```
# Setup
set.seed(1)
n <- 1000
x1<-x2<-x3<-x4<-seq(0,1,by=1)
grid <- expand.grid(x1,x2,x3,x4)

# Calculate probabilities
p <- rep(NA,16)
for (itr in 1:16){
  p[itr] <- exp(grid[itr,1]*grid[itr,2]+grid[itr,2]*grid[itr,3]+grid[itr,3]*grid[itr,4])
}

# Calculate intercept and adjust probabilities
beta_0 <-log(1/sum(p))
for (itr in 1:16){
  p[itr] <- p[itr] * exp(beta_0)
}

# Sample data and fit GLM model
samp <- sample(1:16,prob=p,size=n,replace=TRUE)
count <- rep(NA,16)
for (itr in 1:16){
  count[itr] <- length(which(samp==itr))
}
data <- cbind(count,grid)
names(data)[2:5] <- c("X1","X2","X3","X4")
model <- glm(count ~ X1 + X2 + X3 + X4 + .*, data = data, family = "poisson")
summary(model)
```

The features deemed significant by a t -test are marked with ***. The results are consistent with the true model.

Table I. Regression summary

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	2.97703	0.17268	17.241	< 2e-16	***
X1	-0.09754	0.19408	-0.503	0.615	
X2	-0.25170	0.20105	-1.252	0.211	
X3	0.16575	0.19517	0.849	0.396	
X4	-0.09327	0.19281	-0.484	0.629	
X1:X2	0.96414	0.16313	5.910	3.42e-09	***
X1:X3	-0.13305	0.17915	-0.743	0.458	
X1:X4	0.14144	0.14858	0.952	0.341	
X2:X3	1.13321	0.18387	6.163	7.13e-10	***
X2:X4	0.21445	0.17296	1.240	0.215	
X3:X4	0.84883	0.16996	4.994	5.90e-07	***

Problem 9 [10 pts.]

Let $X_1, \dots, X_n \in \mathbb{R}^d$. Let Σ be the $d \times d$ covariance matrix for X_i . The covariance graph G puts an edge between (j, k) if $\Sigma_{jk} \neq 0$. Here we will use the bootstrap to estimate the covariance graph.

Let Σ have the following form: $\Sigma_{jj} = 1$, $\Sigma_{j,k} = a$ if $|j - k| = 1$ and $\Sigma_{j,k} = 0$ otherwise. Here, $a = 1/4$.

Let $d = 100$ and $n = 50$. Generate n observations. Compute a 95 percent bootstrap confidence set for Σ using the bootstrap distribution

$$\mathbb{P}\left(\max_{j,k} \sqrt{n}|\hat{\Sigma}_{jk}^* - \hat{\Sigma}_{jk}| \leq t \mid X_1, \dots, X_n\right).$$

This gives (uniform) confidence intervals for all the elements of Σ_{jk} . For each (j, k) , put an edge if the confidence interval for Σ_{jk} excludes 0. Plot your graph. Try this for different values of a . Summarize your results.

Solution.

An outline on how to approach this problem is given at page 8 of the Graphical Models notes. In this solution we will use the R package `igraph` to visualize the covariance graph after the estimate.

```
require(mvtnorm)
require(igraph)

set.seed(7)

# Generate Data
n <- 50
d <- 100
sigma_mat <- toeplitz(c(1, 1/4, numeric(d-2)))
data <- rmvnorm(n, numeric(d), sigma_mat)
corr_data <- cor(data)

# Run Bootstrap
bootstrap_rep <- 1e+03
stats_boot <- numeric(bootstrap_rep)
for (b in 1:bootstrap_rep) {
  data_boot <- data[sample(1:n, replace=TRUE),]
  corr_data_boot <- cor(data_boot)
  stats_boot[b] = max(abs(corr_data_boot - corr_data))
}

# Calculate Confidence Sets
alpha_cutoff <- quantile(stats_boot, c(0.05))
corr_low <- corr_data - alpha_cutoff
corr_up <- corr_data + alpha_cutoff

# Remove Self-loops
adjacency_mat <- (corr_low > 0 | corr_up < 0)
for(i in 1:d){
  adjacency_mat[i,i] = 0
}

# Calculate how many are correctly and wrongly recovered
adj_indeces <- which(adjacency_mat == 1, arr.ind = TRUE)
adj_diff_abs <- abs(apply(X=adj_indeces, MARGIN = 1, FUN = diff))
sum(adj_diff_abs == 1)
sum(adj_diff_abs > 1)
```

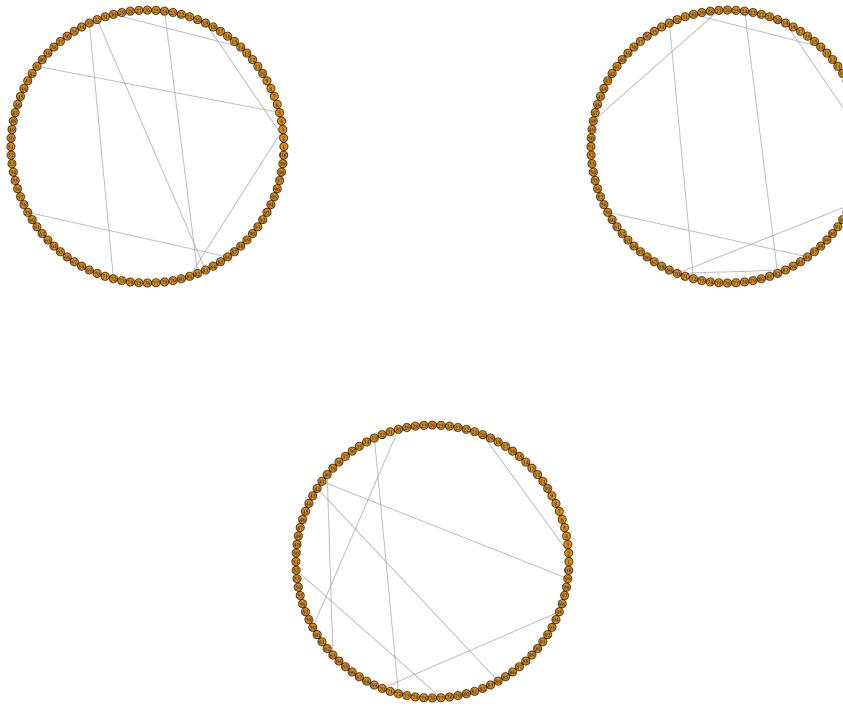


Figure 1: Covariance graph with $a = 1/8, 1/4$ and $1/2$ from top left to center bottom.

```
# Plotting results
colnames(adjacency_mat) <- rownames(adjacency_mat) <- 1:d
graph <- graph_from_adjacency_matrix(adjacency_mat, mode = "undirected", diag = FALSE)
plot(graph, layout = layout.circle, vertex.size=6, vertex.label.cex=0.6)
```

With $a = 1/4$, we recover only 10 out of the 198 non-zero correlations - excluding the diagonal entries - and wrongly recovering 18 of them. When $a = 1/8$ the number of correctly recovered drops to 2, with the wrongly recovered remaining at 16. When further simulating with different values of lower a , reducing a seems to be in general leading to worse performance. When $a = 1/2$ the number of correctly recovered is 150, with the number of mis-recovered equal to 22. Again using simulations with values of a close to $1/2$, recovery performance seem to be improving in this case. It has to be noted that a cannot be larger than $1/2$ as the covariance matrix in that case would not be positive definite anymore.

Problem 10 [8 pts.]

Let $A \in \{0, 1\}$ be a binary treatment variable and let $Y \in \mathbb{R}$ be the response variable. Let $(Y(0), Y(1))$ be the counterfactual variables where $Y = AY(1) + (1 - A)Y(0)$. Assume that

$$Y = \alpha + \gamma A + \sum_{j=1}^d \beta_j X_j + \epsilon$$

where (X_1, \dots, X_d) are confounding variables and $\mathbb{E}[\epsilon | X_1, \dots, X_d] = 0$. Assume there are no unmeasured variables.

- (a) (4 pts.) Let $\theta = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)]$. Show that $\theta = \gamma$.
- (b) (4 pts.) Suppose now that we do not observe the confounding variables X_j . All we observe is $(A_1, Y_1), \dots, (A_n, Y_n)$. Suppose, unaware of the confounding variables, we fit the linear model $Y = \alpha + \rho A + \delta$ where $\mathbb{E}[\delta] = 0$. Let $\hat{\rho}$ be the least squares estimator. Show that $\hat{\rho} \xrightarrow{P} \gamma + \Delta$ for some Δ . Find an explicit expression for Δ .

Solution.

(a)

$$\begin{aligned} \theta &= \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)] \\ &= \mathbb{E}[\mathbb{E}[Y(1)|X_1, \dots, X_n]] - \mathbb{E}[\mathbb{E}[Y(0)|X_1, \dots, X_n]] \\ &= \mathbb{E}\left[\mathbb{E}\left[\alpha + \gamma + \sum_{j=1}^d \beta_j X_j + \epsilon \mid X_1, \dots, X_n\right]\right] - \mathbb{E}\left[\mathbb{E}\left[\alpha + \sum_{j=1}^d \beta_j X_j + \epsilon \mid X_1, \dots, X_n\right]\right] \\ &= \mathbb{E}\left[\alpha + \gamma + \sum_{j=1}^d \beta_j X_j + \mathbb{E}[\epsilon \mid X_1, \dots, X_n]\right] - \mathbb{E}\left[\alpha + \sum_{j=1}^d \beta_j X_j + \mathbb{E}[\epsilon \mid X_1, \dots, X_n]\right] \\ &= \alpha + \gamma + \sum_{j=1}^d \beta_j X_j - \left(\alpha + \sum_{j=1}^d \beta_j X_j\right) \\ &= \gamma \end{aligned}$$

(b)

$$\begin{aligned} \hat{\rho} &= \frac{\frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})(A_i - \bar{A})}{\frac{1}{n} \sum_{i=1}^n (A_i - \bar{A})^2} \\ &\xrightarrow{P} \frac{\text{Cov}(Y, A)}{\text{Var}(A)} && \text{WLLN + conv. thm} \\ &= \frac{\text{Cov}(\alpha + \gamma A + \sum_{j=1}^d \beta_j X_j + \epsilon, A)}{\text{Var}(A)} \\ &= \frac{\gamma \text{Var}(A) + \sum_{j=1}^d \beta_j \text{Cov}(X_j, A) + \text{Cov}(\epsilon, A)}{\text{Var}(A)} \\ &= \gamma + \frac{\sum_{j=1}^d \beta_j \text{Cov}(X_j, A) + \text{Cov}(\epsilon, A)}{\text{Var}(A)} \end{aligned}$$

If we assume $\text{Cov}(\epsilon, A) = 0$ then

$$\Delta = \frac{\sum_{j=1}^d \beta_j \text{Cov}(X_j, A)}{\text{Var}(A)}.$$

Problem 11 [8 pts.]

Consider a sequence of time ordered random variables

$$X_1, A_1, Y_1, X_2, A_2, Y_2, X_3, A_3, Y_3, \dots, X_T, A_T, Y_T.$$

Here, the X'_j 's are the covariates, the A'_j 's are binary treatment variables and the Y'_j 's are the response of interest. Assume there are no unobserved confounding variables. The DAG for this model has all directed arrows from the past into the future. That is, the parents for each variables are all variables in its past. For example, the parent of A_1 is X_1 . The parents of Y_1 are (X_1, A_1) . The parents of X_2 are (X_1, A_1, Y_1) and so on. Let p denote the joint density of all these variables.

- (a) (5 pts.) Find an explicit expression (in terms of p) for

$$\mathbb{E}[Y_T | A_1 = a_1, \dots, A_T = a_T].$$

- (b) (3 pts.) Find an explicit expression (in terms of p) for

$$\mathbb{E}[Y_T | \text{set } (A_1 = a_1, A_2 = a_2, \dots, A_T = a_T)].$$

Solution.

- (a) Let $\text{par}(x_j)$ denote the set of parents of X_j on the DAG, and so on.

$$\begin{aligned} \mathbb{E}[Y_T | A_1 = a_1, \dots, A_T = a_T] &= \int y_T p(y_T | A_1 = a_1, \dots, A_T = a_T) dy_T \\ &= \int y_T \frac{p(y_T, a_1, \dots, a_T)}{p(a_1, \dots, a_T)} dy_T \\ &= \int y_T \frac{\int \cdots \int p(x_1, a_1, y_1, \dots, x_T, a_T, y_T) dx_1 \cdots dx_T dy_1 \cdots dy_{T-1}}{\int \cdots \int p(x_1, a_1, y_1, \dots, x_T, a_T, y_T) dx_1 \cdots dx_T dy_1 \cdots dy_T} dy_T \\ &= \int y_T \frac{\int \cdots \int \prod_j p(x_j | \text{par}(x_j)) p(a_j | \text{par}(a_j)) p(y_j | \text{par}(y_j)) dx_1 \cdots dx_T dy_1 \cdots dy_{T-1}}{\int \cdots \int \prod_j p(x_j | \text{par}(x_j)) p(a_j | \text{par}(a_j)) p(y_j | \text{par}(y_j)) dx_1 \cdots dx_T dy_1 \cdots dy_T} dy_T \\ &= \frac{\int y_T \int \cdots \int \prod_j p(x_j | \text{par}(x_j)) p(a_j | \text{par}(a_j)) p(y_j | \text{par}(y_j)) dx_1 \cdots dx_T dy_1 \cdots dy_{T-1} dy_T}{\int \cdots \int \prod_j p(x_j | \text{par}(x_j)) p(a_j | \text{par}(a_j)) p(y_j | \text{par}(y_j)) dx_1 \cdots dx_T dy_1 \cdots dy_T} \end{aligned}$$

- (b) Using the expression from part (a), we can set $A_1 = a_1, A_2 = a_2, \dots, A_T = a_T$ by replacing $p(a_j | \text{par}(a_j))$ for all $j = 1, \dots, T$ with 1. That is,

$$\begin{aligned} \mathbb{E}[Y_T | \text{set } (A_1 = a_1, A_2 = a_2, \dots, A_T = a_T)] &= \frac{\int y_T \int \cdots \int \prod_j p(x_j | \text{par}(x_j)) p(y_j | \text{par}(y_j)) dx_1 \cdots dx_T dy_1 \cdots dy_{T-1} dy_T}{\underbrace{\int \cdots \int \prod_j p(x_j | \text{par}(x_j)) p(y_j | \text{par}(y_j)) dx_1 \cdots dx_T dy_1 \cdots dy_T}_1} \\ &= \int y_T \left(\int \cdots \int \prod_j p(x_j | \text{par}(x_j)) p(y_j | \text{par}(y_j)) dx_1 \cdots dx_T dy_1 \cdots dy_{T-1} \right) dy_T \end{aligned}$$

Appendix

Theorem 1 Suppose that $X = (X(1), \dots, X(d)) \sim N_d(\mu, \Sigma)$. For any $X_1 \subset X$ and $X_2 = X \setminus X_1$, $X_2|X_1$ follows a multivariate normal distribution.

Proof. We were told we can take this theorem for granted. The parameters that characterize this multivariate normal distribution are computed in Problem 3(a).

Theorem 2 Suppose that $X = (X(1), \dots, X(d)) \sim N_d(\mu, \Sigma)$. For any full-rank $\mathbf{A} \in \mathbb{R}^{m \times d}$ and $\mathbf{b} \in \mathbb{R}^m$,

$$\mathbf{AX} + \mathbf{b} \sim N_m(\mathbf{A}\mu + \mathbf{b}, \mathbf{A}\Sigma\mathbf{A}^T).$$

Proof. We use moment generating functions. Let $Y = \mathbf{AX} + \mathbf{b}$. The joint moment generating function of X is

$$M_X(t) = \exp\left(t^T \mu + \frac{1}{2} t^T \Sigma t\right).$$

Then the joint moment generating function of Y is

$$\begin{aligned} M_Y(t) &= \exp(t^T \mathbf{b}) M_X(\mathbf{A}^T t) \\ &= \exp(t^T \mathbf{b}) \exp\left(t^T \mathbf{A}\mu + \frac{1}{2} t^T \mathbf{A}\Sigma\mathbf{A}^T t\right) \\ &= \exp\left(t^T (\mathbf{A}\mu + \mathbf{b}) + \frac{1}{2} t^T \mathbf{A}\Sigma\mathbf{A}^T t\right), \end{aligned}$$

which is the moment generating function of the joint multivariate normal distribution

$$N_m(\mathbf{A}\mu + \mathbf{b}, \mathbf{A}\Sigma\mathbf{A}^T). \blacksquare$$

Corollary 3 Suppose that $X = (X(1), \dots, X(d)) \sim N_d(\mu, \Sigma)$. Then any p -dimensional subset \tilde{X} of X follows a multivariate normal distribution

$$\tilde{X} \sim N_p(\tilde{\mu}, \tilde{\Sigma}),$$

where $\tilde{\mu}$ is the vector of means of the variables in $\tilde{X} \subseteq X$ and $\tilde{\Sigma}$ is the sub-matrix of Σ obtained by deleting the rows and columns corresponding to the variables in $X \setminus \tilde{X}$.

Theorem 4 Suppose that $X = (X(1), \dots, X(d)) \sim N_d(\mu, \Sigma)$. Two random-vectors $\tilde{\mathbf{X}}_1 \subset X$ and $\tilde{\mathbf{X}}_2 \subset X$ are independent if and only if they are uncorrelated.

Proof.

“ \implies ” This is true regardless of the distribution. See [?].

“ \impliedby ” By Corollary 2, $\tilde{X} = (\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2) \in \mathbb{R}^q = \mathbb{R}^{r+s}$ follows a multivariate normal distribution with density

$$f_{\tilde{X}}(\tilde{x}_1, \dots, \tilde{x}_q) = \frac{1}{\sqrt{(2\pi)^q |\tilde{\Sigma}|}} \exp\left(-\frac{1}{2} (\tilde{\mathbf{x}} - \tilde{\mu})^T \tilde{\Sigma}^{-1} (\tilde{\mathbf{x}} - \tilde{\mu})\right), \quad (3)$$

with

$$\tilde{\mu} = \begin{pmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_2 \end{pmatrix} \quad \text{and} \quad \tilde{\Sigma} = \begin{pmatrix} \tilde{\Sigma}_{11} & 0 \\ 0 & \tilde{\Sigma}_{22} \end{pmatrix},$$

where

$$\tilde{\mathbf{X}}_1 \sim N_r(\tilde{\boldsymbol{\mu}}_1, \tilde{\Sigma}_{11}) \quad \text{and} \quad \tilde{\mathbf{X}}_2 \sim N_s(\tilde{\boldsymbol{\mu}}_2, \tilde{\Sigma}_{22}).$$

In the exponent of (3) we have

$$\begin{aligned} & (\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}})^T \tilde{\Sigma}^{-1} (\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}}) \\ &= (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1, \tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2)^T \left(\begin{array}{c|c} \tilde{\Sigma}_{11} & 0 \\ \hline 0 & \tilde{\Sigma}_{22} \end{array} \right)^{-1} (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1, \tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2) \\ &= (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1, \tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2)^T \left(\begin{array}{c|c} \tilde{\Sigma}_{11}^{-1} & 0 \\ \hline 0 & \tilde{\Sigma}_{22}^{-1} \end{array} \right) (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1, \tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2) \\ &= (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1)^T \tilde{\Sigma}_{11} (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1) + (\tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2)^T \tilde{\Sigma}_{22} (\tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2). \end{aligned}$$

Hence, (3) factorizes as follows

$$\begin{aligned} & f_{\tilde{X}}(\tilde{x}_1, \dots, \tilde{x}_q) \\ &= \frac{1}{\sqrt{(2\pi)^q |\tilde{\Sigma}|}} \exp \left(-\frac{1}{2} (\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}})^T \tilde{\Sigma}^{-1} (\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}}) \right) \\ &= \frac{1}{\sqrt{(2\pi)^{r+s} |\tilde{\Sigma}_{11}| |\tilde{\Sigma}_{22}|}} \exp \left(-\frac{1}{2} (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1)^T \tilde{\Sigma}_{11} (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1) + (\tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2)^T \tilde{\Sigma}_{22} (\tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2) \right) \\ &= \frac{1}{\sqrt{(2\pi)^r |\tilde{\Sigma}_{11}|}} \exp \left(-\frac{1}{2} (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1)^T \tilde{\Sigma}_{11} (\tilde{\mathbf{x}}_1 - \tilde{\boldsymbol{\mu}}_1) \right) \cdot \frac{1}{\sqrt{(2\pi)^s |\tilde{\Sigma}_{22}|}} \exp \left(-\frac{1}{2} (\tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2)^T \tilde{\Sigma}_{22} (\tilde{\mathbf{x}}_2 - \tilde{\boldsymbol{\mu}}_2) \right) \\ &= f_{\tilde{\mathbf{X}}_1}(\tilde{\mathbf{x}}_1) f_{\tilde{\mathbf{X}}_2}(\tilde{\mathbf{x}}_2). \blacksquare \end{aligned}$$

36-708 Statistical Machine Learning Homework #4 Solutions

DUE: April 19, 2019

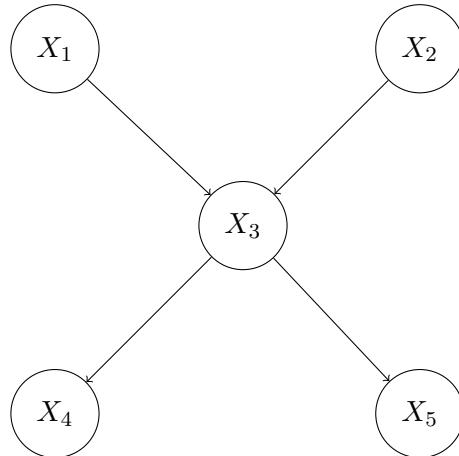
Problem 1 [5 pts.]

Consider the directed graph with vertices $V = \{X_1, X_2, X_3, X_4, X_5\}$ and edge set $E = \{(1,3), (2,3), (3,4), (3,5)\}$.

- (a) **[2 pts.]** List all the independence statements implied by this graph.
- (b) **[1 pts.]** Find the causal distribution $p(x_4|\text{set } x_3 = s)$.
- (c) **[2 pts.]** Find the implied undirected graph for these random variables. Which independence statements get lost in the undirected graph (if any)?

Solution.

The graph can be visualized as follows:



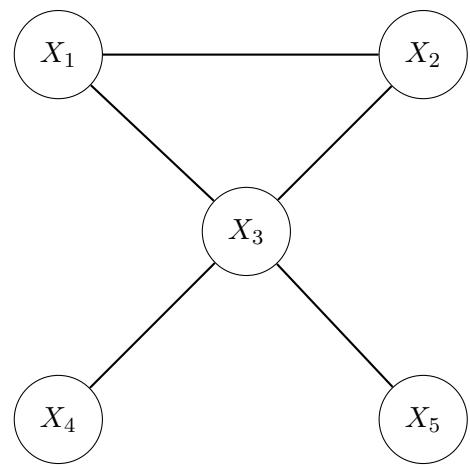
- (a) The independence statements implied are the following:

- $X_1 \perp\!\!\!\perp X_2$;
- $X_4 \perp\!\!\!\perp \{X_1, X_2\} | X_3$ and $X_5 \perp\!\!\!\perp \{X_1, X_2\} | X_3$;
- $X_5 \perp\!\!\!\perp X_4 | X_3$.

- (b) Given that we set $X_3 = X_3$, then by the independence highlighted above X_1 and X_2 can be dropped from the graph. Hence we have that:

$$p(x_4|\text{set } x_3 = s) = \int p_*(x_4, x_5) dx_5 = \int p(x_4|x_3 = s)p(x_5|x_3 = s) dx_5 = p(x_4|x_3 = s)$$

- (c) The moralized graph becomes:



We lose the (unconditional) independence between X_1 and X_2 during the moralization process, while all the others are retained.

Problem 2 [20 pts.]

Let $d \geq 2$, and let $X_1, \dots, X_n \sim P$ where $X_i = (X_i(1), \dots, X_i(d)) \in \mathbb{R}^d$. Assume that the coordinates of X_i are independent. Further, assume that $X_i(j) \sim \text{Bernoulli}(p_j)$ where $0 < c \leq p_j \leq C < 1$. Let \mathcal{P} be all such distributions. Let

$$R_n = \inf_{\hat{p}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \|\hat{p} - p\|_\infty.$$

Find lower and upper bounds on the minimax risk.

Solution.

For **upper bound**, consider an estimator \bar{X} for estimating p . Then $\bar{X} - p$ is sub-Gaussian with parameter $\sigma^2 = \frac{1}{4n}$. Hence, from the lemma below,

$$\mathbb{E} \|\bar{X} - p\|_\infty = \mathbb{E} \left[\max_{1 \leq i \leq d} |\bar{X} - p|_i \right] \leq \frac{1}{2\sqrt{n}} \sqrt{2 \log(2d)}.$$

And since $d \geq 2$, we have $2 \log(2d) \leq 4 \log d$, so,

$$\inf_{\hat{p}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \|\hat{p} - p\|_\infty \leq \mathbb{E} \|\bar{X} - p\|_\infty \leq \sqrt{\frac{\log d}{n}}.$$

For **lower bound**, let $\alpha = \sqrt{\frac{\log d}{16n}}$, $p^{(0)} = (c, \dots, c) \in \mathbb{R}^d$.

And for $1 \leq j \leq d$, construct d -dimensional vector parameter

$$p_j := (p_j^{(i)}, i = 1, \dots, d) = (c, \dots, c, \underbrace{c + \alpha}_{j\text{th}}, c, \dots, c) \in \mathbb{R}^d,$$

then $\|p_j - p_k\|_\infty = \alpha$ for $j \neq k$, so

$$\min_{j \neq k} \|p_j - p_k\|_\infty = \alpha.$$

Let P_j be the multivariate Bernoulli with parameter p_j , then it's a product of uni-variate Bernoulli's, $P_j = \prod_{i=1}^d P_j^{(i)}$, where $P_j^{(i)}$ is uni-variate Bernoulli with parameter $p_j^{(i)}$. Then

$$KL(P^{(j)}, P^{(k)}) = \sum_{i=1}^d KL(P_i^{(j)}, P_i^{(k)}) = KL(P_j^{(j)}, P_j^{(k)}) + KL(P_k^{(j)}, P_k^{(k)}).$$

since they only differ in two terms: $i = j$ or $i = k$. Let uni-variate Beroulli with parameter c as P_c , then,

$$\begin{aligned} KL(P^{(j)}, P^{(k)}) &= KL(P_j^{(j)}, P_j^{(k)}) + KL(P_k^{(j)}, P_k^{(k)}) \\ &= \sum_x P_{c+\alpha} \log\left(\frac{P_{c+\alpha}}{P_c}\right) + \sum_x P_c \log\left(\frac{P_c}{P_{c+\alpha}}\right) \\ &= \sum_x (P_{c+\alpha} - P_c) \log\left(\frac{P_{c+\alpha}}{P_c}\right) \\ &= \alpha \log \frac{(\alpha + c)(1 - c)}{(1 - \alpha - c)c} \\ &\leq C\alpha^2, \quad \text{by Taylor expansion.} \end{aligned}$$

(Fano's method)

$$\begin{aligned}\max_{j \neq k} KL(P_j, P_k) &= \max_{j \neq k} \frac{1}{2} \|\mu_j - \mu_k\|_2^2 = C\alpha^2 \\ &= \frac{\log d}{8n} \leq \frac{\log(d+1)}{4n}.\end{aligned}$$

Hence by Corollary 13 in the minimax notes where $N = d + 1$,

$$\inf_{\hat{p}} \sup_{P \in \mathcal{P}} \mathbb{E}_P \|\hat{p} - p\|_\infty \geq \frac{\alpha}{4} = \sqrt{\frac{\log d}{256n}}.$$

Note: following the same construction of $p^{(i)}$ and KL distance bound, lower bounds for minimax with the same rate with respect to d and n can also be derived using Theorem 12 or Theorem 14.

Lemma 1 (*Maximal inequality for subgaussian random variables*)
Let $\{X_i\}_{1 \leq i \leq n}$ be sub-Gaussian variables with parameter σ^2 , then

$$\mathbb{E} \left[\max_{1 \leq i \leq n} X_i \right] \leq \sigma \sqrt{2 \log n} \text{ and } \mathbb{E} \left[\max_{1 \leq i \leq n} |X_i| \right] \leq \sigma \sqrt{2 \log(2n)}.$$

It's covered in Advanced stats.

See http://www.stat.cmu.edu/~arinaldo/Teaching/36755/F17/Scribed_Lectures/F17_0911.pdf.

Problem 3 [20 pts.]

Let $\{p_\theta : \theta \in \Theta\}$ where $\Theta \subset \mathbb{R}$ be a parametric model. Suppose that the model satisfies the usual regularity conditions. In particular, the Fisher information $I(\theta)$ is positive and smooth and the mle has the usual nice properties. Let the loss function be $L(\hat{\theta}, \theta) = H(p_{\hat{\theta}}, p_\theta)$ where H denotes Hellinger distance. Find the minimax rate.

Solution.

Assume that the densities in $\{p_\theta : \theta \in \Theta\}$ is differentiable in quadratic mean (QMD) as equation (27) in the Minimax note, which says the Hellinger distance between $p_{\theta+h}, p_\theta$ can be approximated by first order Taylor expansion on h . In other words, the loss with Hellinger distance is then approximately equivalent with that with squared loss on θ ,

$$H^2(p_{\theta+h}, p_\theta) = \frac{1}{8} \|h\|^2 I(\theta) + o(\|h\|^2).$$

Proof see <https://www.stat.berkeley.edu/~bartlett/courses/2013spring-stat210b/notes/25notes.pdf>. So the minimax rate for Hellinger loss is the same as for squared risk $R(\hat{\theta}, \theta) = \|\hat{\theta} - \theta\|^2$,

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta} H^2(p_{\hat{\theta}}, p_\theta) = O(\inf_{\hat{\theta}} \sup_{\theta \in \Theta} R(\hat{\theta}_n^{mle}, \theta)).$$

Next, we prove that the minimax rate for squared loss is achieved by MLE estimator. For a fixed true parameter θ , denote the MLE as $\hat{\theta}_n^{mle}$ from sample $X_1, \dots, X_n \sim P_\theta$. By MLE asymptotic distribution,

$$\sqrt{n}(\hat{\theta}_n^{mle} - \theta) \rightarrow N(0, I^{-1}(\theta)).$$

So the squared risk for the MLE is,

$$R(\hat{\theta}_n^{mle}, \theta) = Var(\hat{\theta}_n^{mle}) + bias^2 \rightarrow I^{-1}(\theta)/n.$$

For any other estimator $\hat{\theta}$, by theorem 17 with $\psi(x) = x$ and l as the square loss, under the QMD condition, the squared risk is lower bounded by that for MLE,

$$R(T_n, \theta) = Var(T_n) + bias^2 \geq Var(T_n) \geq Var(U) = I^{-1}(\theta)/n = R(\hat{\theta}_n^{mle}, \theta).$$

This lower bound holds for all $\theta \in \Theta$. So the minimax risk for squared loss is achieved by the MLE,

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta} R(\hat{\theta}, \theta) = \sup_{\theta \in \Theta} R(\hat{\theta}_n^{mle}, \theta) \rightarrow \frac{1}{n} \sup_{\theta \in \Theta} I^{-1}(\theta).$$

Therefore,

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta} H^2(p_{\hat{\theta}}, p_\theta) = O\left(\frac{1}{n}\right),$$

or equivalently,

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta} H(p_{\hat{\theta}}, p_\theta) = O(n^{-1/2}).$$

Alternative solution (thank Tim Barry for the ideas)

We first derive an upper bound for the KL distance between arbitrary $p_{\theta+h}, p_\theta$,

$$\begin{aligned}
 KL(p_{\theta+h}, p_\theta) &= \int \log \frac{p_\theta}{p_{\theta+h}} p_\theta dx \\
 &= \int \log p_\theta - \log p_{\theta+h} p_\theta dx \\
 &= \int \log p_\theta - (\log p_\theta + h \frac{\partial \log p_\theta}{\partial \theta} + h^2 \frac{\partial^2 \log p_\theta}{\partial \theta^2} + o(h^2)) p_\theta dx \\
 &= h^2 \left(- \int \frac{\partial^2 \log p_\theta}{\partial \theta^2} p_\theta dx \right) + o(h^2) \\
 &\leq CI(\theta)h^2,
 \end{aligned}$$

where we assume good properties for the density to allow swapping the derivative and integral, so that,

$$\int \frac{\partial \log p_\theta}{\partial \theta} p_\theta dx = \int \frac{1}{p_\theta} \frac{\partial p_\theta}{\partial \theta} p_\theta dx = \int \frac{\partial p_\theta}{\partial \theta} dx = \frac{\partial (\int p_\theta dx)}{\partial \theta} = 0.$$

For **upper bound**, consider MLE estimator,

$$H(p_{\hat{\theta}^{mle}}, p_\theta) \leq \sqrt{KL(p_{\hat{\theta}^{mle}}, p_\theta)} \leq \sqrt{C(\hat{\theta}^{mle} - \theta)^2 I(\theta)},$$

By MLE asymptotic distribution,

$$\hat{\theta}^{mle} - \theta \rightarrow N(0, I^{-1}(\theta)/n),$$

thus,

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta} H(p_{\hat{\theta}}, p_\theta) \leq H(p_{\hat{\theta}^{mle}}, p_\theta) \leq \sqrt{CI(\theta)(Var(\hat{\theta}_n^{mle}) + bias^2)} \rightarrow O(n^{-1/2}).$$

For **lower bound**, consider two distribution p_θ and $p_{\theta+h}$, where $h = \sqrt{\frac{\log 2}{CI(\theta)n}}$, then by QMD condition,

$$H(p_{\theta+h}, p_\theta) = Ch = O(n^{-1/2}).$$

And the KL distance,

$$KL(p_{\theta+h}, p_\theta) \leq CI(\theta)h^2 \leq \frac{\log 2}{n}.$$

Hence by Corollary 5 in the minimax notes,

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta} H(p_{\hat{\theta}}, p_\theta) = O(n^{-1/2}).$$

Problem 4 [15 pts.]

Let $Y = (Y_1, \dots, Y_d) \sim N(\theta, I)$ where $\theta = (\theta_1, \dots, \theta_d)$. Assume that $\theta \in \Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq 1\}$. Let

$$R_d = \inf_{\widehat{\theta}} \sup_{\theta \in \Theta} \mathbb{E}_{\theta} \|\widehat{\theta} - \theta\|^2.$$

Show that $c \log d \leq R_d \leq C \log d$ for some constants c and C .

Solution.

For the upper bound, we first prove a high-probability bound lemma for the maximum of Gaussians (from 36-705, Lecture 27, Fall 2018):

Lemma 2 Suppose that, $\epsilon_1, \dots, \epsilon_d \sim N(0, \sigma^2)$ then with probability at least $1 - \delta$,

$$\max_{i=1}^d |\epsilon_i| \leq \sigma \sqrt{2 \log(2d/\delta)}.$$

Proof. By the Gaussian tail bound, if $\epsilon \sim N(0, \sigma^2)$:

$$\mathbb{P}(|\epsilon| \geq t) \leq 2 \exp(-t^2/(2\sigma^2)),$$

By using the union bound:

$$\mathbb{P}(\max_i |\epsilon_i| \geq t) \leq 2d \exp(-t^2/(2\sigma^2)),$$

By setting $2d \exp(-t^2/(2\sigma^2)) = \delta$ we obtain the lemma.

Now, we assume $\widehat{\theta}$ to be the hard-thresholding estimator, defined as:

$$\widehat{\theta}_i = y_i \mathbb{I}(|y_i| \geq t), \quad \forall i \in \{1, \dots, d\},$$

We have the following theorem (from 36-705, Lecture 27, Fall 2018):

Theorem 3 Suppose we choose the threshold:

$$t = 2\sigma \sqrt{2 \log(2d/\delta)},$$

then with probability at least $1 - \delta$,

$$\|\widehat{\theta} - \theta\|_2^2 \leq 9 \sum_{i=1}^d \min \left\{ \theta_i^2, \frac{t^2}{4} \right\} \leq Ct^2$$

For some $C_1 > 0 \in \mathbb{R}$.

Proof. We condition on the event from the previous lemma, i.e. that

$$\max_{i=1}^d |\epsilon_i| \leq \sigma \sqrt{2 \log(2d/\delta)} \leq \frac{t}{2}.$$

Now, observe that,

$$\|\widehat{\theta} - \theta\|_2^2 = \sum_{i=1}^d (\widehat{\theta}_i - \theta_i)^2,$$

so we can consider each co-ordinate separately. Let us consider some cases:

1. If for any co-ordinate $|\theta_i| \leq \frac{t}{2}$ our estimate is 0, so our risk for that coordinate is simply θ_i^2 .
2. If $|\theta_i| \geq \frac{3t}{2}$ our estimate is simply $\widehat{\theta}_i = y_i$ so our risk is simply $\epsilon_i^2 \leq \frac{t^2}{4}$.
3. If $\frac{t}{2} \leq |\theta_i| \leq \frac{3t}{2}$, then our risk,

$$(\widehat{\theta}_i - \theta_i)^2 = (y_i \mathbb{I}(|y_i| \geq t) - \theta_i)^2 = \theta_i^2 \mathbb{I}(|y_i| < t) + \epsilon_i^2 \mathbb{I}(|y_i| \geq t) \leq \max\{\epsilon_i^2, \theta_i^2\} \leq \frac{9t^2}{4}.$$

Putting these together we see that,

$$\|\widehat{\theta} - \theta\|_2^2 \leq 9 \sum_{i=1}^d \min \left\{ \theta_i^2, \frac{t^2}{4} \right\} = 9 \sum_{\theta_i=0} \min \left\{ \theta_i^2, \frac{t^2}{4} \right\} + 9 \sum_{\theta_i \neq 0} \min \left\{ \theta_i^2, \frac{t^2}{4} \right\} \leq C_1 t^2$$

We also have that, under the same assumptions of the theorems above:

$$\|\widehat{\theta} - \theta\|_2^2 = \sum_{i=1}^d (y_i \mathbb{I}(|y_i| \geq t) - \theta_i)^2 \leq C_2 d \max_{i=1,\dots,d} (t - \theta_i)^2 \leq C_2 d t^2$$

And so we have that:

$$\begin{aligned} \mathbb{E} [\|\widehat{\theta} - \theta\|_2^2] &= \int_0^\infty \mathbb{P} (\|\widehat{\theta} - \theta\|_2^2 > x) dx \\ &= \int_0^{C_1 t^2} \mathbb{P} (\|\widehat{\theta} - \theta\|_2^2 > x) dx + \int_{C_1 t^2}^{C_2 d t^2} \mathbb{P} (\|\widehat{\theta} - \theta\|_2^2 > x) dx \\ &\leq \int_0^{C_1 t^2} (1 - \delta) dx + \int_{C_1 t^2}^{C_2 d t^2} \delta dx \\ &\leq C_1 t^2 (1 - \delta) + t^2 (C_2 d - 1) \delta \leq C_3 t^2 \approx C_3 \log d \end{aligned}$$

For the lower bound, let $P_j = N(\theta_j, I)$ where $\theta_0 = (0, \dots, 0)$ and θ_j is the d -dimensional vector ($d \geq 8$) where

$$\theta_j(k) = \begin{cases} \sqrt{\frac{\log d}{32}} & k = j \\ 0 & k \neq j. \end{cases}$$

P_0 is absolutely continuous wrt each other distribution and for all $j = 1, \dots, d$, we claim

$$KL(P_j, P_0) \leq \frac{\log d}{16}.$$

The statement above actually works for any P_j, P_i with $i \neq j$. Let $P_i = N(\mu_i, I)$, then:

$$\begin{aligned} KL(P_j, P_k) &= \int \frac{1}{2} (\|x - \mu_k\|_2^2 - \|x - \mu_j\|_2^2) P_j(x) dx \\ &= \frac{1}{2} \|\mu_j - \mu_k\|_2^2. \end{aligned}$$

Which implies:

$$\max_{j \neq k} KL(P_j, P_k) = \max_{j \neq k} \frac{1}{2} \|\mu_j - \mu_k\|_2^2 = 2 \left(\sqrt{\frac{\log d}{32}} \right)^2 = \frac{\log d}{16}$$

It then follows that:

$$\frac{1}{d} \sum_{j=1}^d KL(P_j, P_0) \leq \frac{\log d}{16}.$$

Therefore, by Tsybakov's bound,

$$\begin{aligned} R_d &\geq \frac{s}{16} \\ &= \frac{\max_{j \neq k} \|\theta_j - \theta_k\|_2^2}{16} \\ &= \frac{2\sqrt{\frac{\log d}{32}}^2}{16} \\ &\geq c \log d. \end{aligned}$$

Problem 5 [20 pts.]

Let $X_1, \dots, X_n \sim F$ where F is some distribution on \mathbb{R} . Suppose we put a Dirichlet process prior on F :

$$F \sim \text{DP}(\alpha, F_0).$$

- (a) **(10 pts.)** Recall the stick-breaking construction. Show that $\mathbb{E}(\sum_{j=1}^{\infty} W_j) = 1$.
- (b) **(10 pts.)** Simulate $n = 10$ data points from a $N(0, 1)$. Try three values of α : namely, $\alpha = .1$, $\alpha = 1$ and $\alpha = 10$. Compute the 95 percent Bayesian confidence band and the 95 percent DKW band. Plot the results for one example. Now repeat the simulation 1,000 times and report the coverage probability for each confidence band.

Solution.

- (a) We start by showing $\mathbb{P}(\sum_{j=1}^{\infty} W_j = 1) = 1$. The expectation will then follow easily. First we prove a series of lemmas.

Lemma 4 *For all $n \in \mathbb{N}$,*

$$1 - \sum_{j=1}^n W_j = \prod_{j=1}^n (1 - V_j).$$

Proof. (by induction)

Base case. $k = 1$

$$1 - W_1 = 1 - V_1. \quad \checkmark$$

Inductive hypothesis. Now assume that

$$1 - \sum_{j=1}^{n-1} W_j = \prod_{j=1}^{n-1} (1 - V_j).$$

Inductive step.

$$\begin{aligned} 1 - \sum_{j=1}^n W_j &= 1 - \sum_{j=1}^{n-1} W_j - W_n \\ &= \prod_{j=1}^{n-1} (1 - V_j) - V_n \prod_{j=1}^{n-1} (1 - V_j) \\ &= (1 - V_n) \prod_{j=1}^{n-1} (1 - V_j) \\ &= \prod_{j=1}^n (1 - V_j). \quad \checkmark \end{aligned} \tag{1}$$

Lemma 5 *Let v_1, v_2, \dots be a sequence such that $0 < v_j < 1$ for all j . Then*

$$\prod_{j=1}^{\infty} (1 - v_j) > 0 \quad \text{if and only if} \quad \sum_{j=1}^{\infty} v_j < \infty.$$

Proof. First notice that

$$-\log \prod_{j=1}^{\infty} (1 - v_j) = -\sum_{j=1}^{\infty} \log(1 - v_j),$$

and thus

$$\prod_{j=1}^{\infty} (1 - v_j) > 0 \iff -\sum_{j=1}^{\infty} \log(1 - v_j) < \infty.$$

We now have

$$\left\{ -\log(1 - v_j) \right\}_{j \in \mathbb{N}} > 0 \quad \text{and} \quad \{v_j\}_{j \in \mathbb{N}} > 0,$$

so we can use the Limit Comparison test to prove that

$$-\sum_{j=1}^{\infty} \log(1 - v_j) < \infty \iff \sum_{j=1}^{\infty} v_j < \infty.$$

Since both series diverge when $v_j \rightarrow 0$, it suffices to consider only the sequences where $v_j \rightarrow 0$.

$$\lim_{v_j \rightarrow 0} \frac{-\log(1 - v_j)}{v_j} \stackrel{L'H}{=} \lim_{v_j \rightarrow 0} \frac{1}{1 - v_j} \quad (2)$$

$$= 1. \quad (3)$$

Hence,

$$-\sum_{j=1}^{\infty} \log(1 - v_j) < \infty \quad \text{if and only if} \quad \sum_{j=1}^{\infty} v_j < \infty,$$

and thus,

$$\prod_{j=1}^{\infty} (1 - v_j) > 0 \quad \text{if and only if} \quad \sum_{j=1}^{\infty} v_j < \infty.$$

Corollary 6 Let v_1, v_2, \dots be a sequence such that $0 < v_j < 1$ for all j . Then

$$\prod_{j=1}^{\infty} (1 - v_j) = 0 \quad \text{if and only if} \quad \sum_{j=1}^{\infty} v_j = \infty.$$

Lemma 7 (Borel-Cantelli) If $\sum_{j=1}^{\infty} \mathbb{P}(V_j > \epsilon) = \infty$ for some $\epsilon > 0$, then $\mathbb{P}(V_j > \epsilon \text{ i.o.}) = 1$.

Now since

$$V_j \sim \text{Beta}(1, \alpha), \quad j = 1, 2, \dots$$

we have

$$\mathbb{P}(V_j > \epsilon) > 0 \quad \text{for all } \epsilon \in (0, 1) \quad \text{and} \quad j = 1, 2, \dots \quad (4)$$

since the beta distribution puts positive mass over its entire support $(0, 1)$, and now (4) implies

$$\sum_{j=1}^{\infty} \mathbb{P}(V_j > \epsilon) = \infty \quad \text{for all } \epsilon \in (0, 1). \quad (5)$$

So altogether,

$$\begin{aligned}
 \mathbb{P}\left(\sum_{j=1}^{\infty} W_j = 1\right) &\stackrel{\text{Lemma 2}}{=} \mathbb{P}\left(\prod_{j=1}^{\infty}(1 - V_j) = 0\right) \\
 &\stackrel{\text{Cor. 4}}{=} \mathbb{P}\left(\sum_{j=1}^{\infty} V_j = \infty\right) \\
 &\geq \mathbb{P}(V_j > \epsilon \text{ i.o.}) \\
 &\stackrel{\text{Lemma 5}}{=} 1,
 \end{aligned} \tag{6}$$

where the inequality comes from the fact

$$\left\{ \sum_{j=1}^{\infty} V_j = \infty \right\} \supset \left\{ V_j > \epsilon \text{ i.o.} \right\}.$$

Thus,

$$\mathbb{P}\left(\sum_{j=1}^{\infty} W_j = 1\right) = 1. \blacksquare$$

It follows that

$$\begin{aligned}
 \mathbb{E}\left(\sum_{j=1}^{\infty} W_j\right) &= \int_{\mathbb{R}} \sum_{j=1}^{\infty} W_j dF \\
 &= \mathbb{P}\left(\sum_{j=1}^{\infty} W_j = 1\right) \cdot 1 + 0 \\
 &= 1,
 \end{aligned}$$

where F is the distribution function of the random variable $\sum_{j=1}^{\infty} W_j$.

- (b) Two good resources for such simulation are the [distr package](#) in R, which is showcased by [this tutorial](#). For Python 3 [this tutorial](#) uses the pyMC3 modules to provide a achieve a similar goal. We include R code for a single simulation, with parts taken from the tutorial indicated above.

```

library(distr)
library(coda)
library(latex2exp)

# Setup
set.seed(7)
n <- 10
alpha_vec <- c(0.1, 1, 10)
x_grid <- seq(-3, 3, by=0.05)
signif_level <- 0.05

# Sample observations
x_pts <- rnorm(n)

# Generate DKW Band
x_ecdf <- ecdf(x_pts)
x_ecdf_error <- sqrt(log(2 / signif_level) / (2 * n))

```

```
dkw.lb <- pmax(x_ecdf(x_grid) - x_ecdf_error, 0)
dkw.ub <- pmin(x_ecdf(x_grid) + x_ecdf_error, 1)

## BAYESIAN BANDS
# Functions to generate Bayesian Credible Bands
sample_cdf <- function(F_hat, n){
  F_hat@r(n) # F_hat is a S6 class object from the distr package
}

# Sampling from the prior distribution
sample_prior <- function(F0, alpha, n){
  cdf_sample <- sample_cdf(F0, n)
  v <- rbeta(n, 1, alpha) # See 5a) for definitions
  w <- c(v[1], rep(0, n-1))
  for(ii in 2:n){
    w[ii] <- v[ii]*cumprod(1-v)[ii-1] # See 5a) for definitions
  }
  function(m){
    sample(cdf_sample, m, prob=w, replace=T)
  }
}

# Sampling from the posterior distribution
sample_posterior <- function(F0, alpha, data){
  n <- length(data)
  F_hat <- DiscreteDistribution(data) #distr function for empirical CDF
  F_for_post <- n/(n+alpha)*F_hat+alpha/(n+alpha)*F0
  sample_prior(F_for_post, alpha+n, n)
}

# Now simulate for all different alphas
list_out_bayes <- list()
for(alpha in alpha_vec){
  iters <- 100
  m <- 1000
  F0 <- DiscreteDistribution(rnorm(m))

  y <- matrix(nrow=length(x_grid), ncol=iters)
  for(iter in 1:iters){
    F_post <- sample_posterior(F0, alpha, x_pts)
    y[,iter] <- ecdf(F_post(m))(x_grid)
  }

  mean_post_sim <- rowMeans(y) #Posterior Mean
  cred_int <- apply(y, 1, function(row) HPDinterval(as.mcmc(row),prob=signif_level))
    #obtains 95% credible interval
  list_out_bayes[[as.character(alpha)]] <- list('cred_int' = cred_int,
    'mean_post_sim'=mean_post_sim)
}

# Plot the results for each of the alpha
for (alpha_val in alpha_vec){
  plot(x_ecdf, xlim=c(-3, 3),
    main = TeX(sprintf("95%% DKW and Bayesian Credible Band ($\\alpha = %s$)", as.character(alpha_val)))) #ECDF
```

```

lines(x_grid, dkw.lb, col="green") #DKW
lines(x_grid, dkw.ub, col="green")

points(x_grid, list_out_bayes[[as.character(alpha_val)]]$'mean_post_sim',
       type='l', col="red")
points(x_grid, list_out_bayes[[as.character(alpha_val)]]$'cred_int'[2,], type='l',
       col='blue')
points(x_grid, list_out_bayes[[as.character(alpha_val)]]$'cred_int'[1,], type='l',
       col='blue')
curve(pnorm, xlim=c(-3, 3), add=TRUE, col="red", lwd=2)
legend("topleft", lty="solid", legend=c("True", "DKW", "Kolmogorov", "Posterior
Mean"),
      col=c("black", "green", "blue", "red"))
}

```

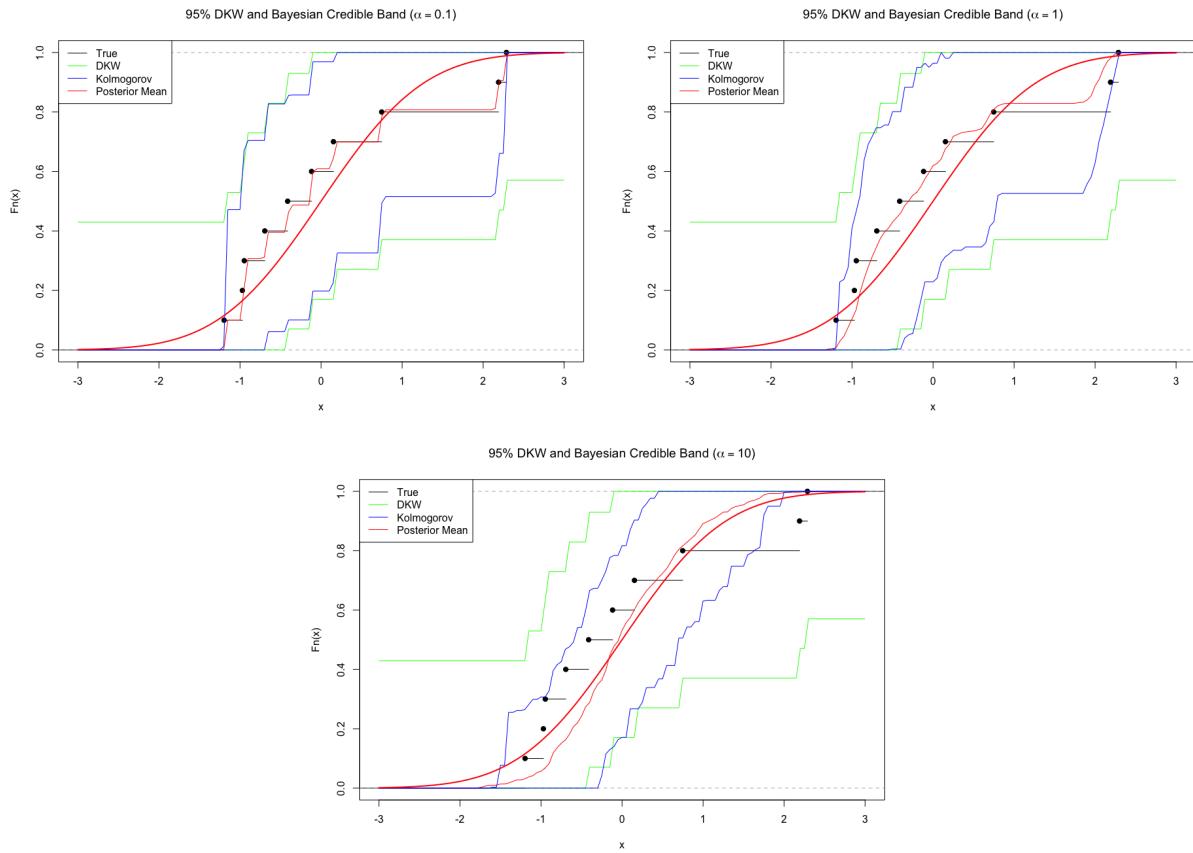


Figure 1: 95% DKW and Bayesian credible bands at different alpha levels: 0.1, 1, 10 from upper left to bottom center respectively

For $n = 1,000$ simulations, one should replicate the above code and consider whether the full empirical CDF is captured between the bands for both the Bayesian confidence and DKW bands.

Problem 6 [20 pts.]

In this question we consider a nonparametric Bayesian estimator and compare to the minimax estimator. For $i = 1, \dots, n$ and $j = 1, 2, \dots$ let

$$X_{ij} = \theta_j + \epsilon_{ij}$$

where all the ϵ'_{ij} s are independent $N(0, 1)$. The parameter is $\theta = (\theta_1, \theta_2, \dots)$. Assume that $\sum_j \theta_j^2 < \infty$. Due to sufficiency, we can reduce the problem to the sample means. Thus let $Y_j = n^{-1} \sum_{i=1}^n X_{ij}$. So the model is $Y_j \sim N(\theta_j, 1/n)$ for $j = 1, 2, 3, \dots$. We will put a prior π on θ as follows. We take each θ_j to be independent and we take $\theta_j \sim N(0, \tau_j^2)$.

(a) (5 pts.) Find the posterior for θ . Find the posterior mean $\widehat{\theta}$.

(b) (7 pts.) Suppose that $\sum_j \tau_j^2 < \infty$. Show that $\widehat{\theta}$ is consistent, that is, $\|\widehat{\theta} - \theta\|^2 \xrightarrow{P} 0$.

(c) (8 pts.) Now suppose that θ is in the Sobolev ball

$$\Theta = \left\{ \theta = (\theta_1, \theta_2, \dots) : \sum_j j^{2p} \theta_j^2 \leq C^2 \right\}$$

where $p > 1/2$. The minimax (for squared error loss) for this problem is $R_n \asymp n^{-2p/(2p+1)}$. Let $\tau_j^2 = (1/j)^{2r}$. Find r so that the posterior mean achieves the minimax rate.

Solution.

(a) By Theorem 6 (in the appendix) we have

$$\widehat{\theta}_j = \frac{n Y_j \tau_j^2}{1 + n \tau_j^2}.$$

(b) For any $\epsilon > 0$,

$$\begin{aligned} P\left(\|\widehat{\theta} - \theta\|^2 > \epsilon\right) &\leq \frac{\mathbb{E}\left[\|\widehat{\theta} - \theta\|^2\right]}{\epsilon} && \text{Markov's inequality} \\ &= \frac{1}{\epsilon} \sum_{j=1}^{\infty} \mathbb{E}\left[(\widehat{\theta}_j - \theta_j)^2\right] \\ &= \frac{1}{\epsilon} \left[\sum_{j=1}^{\infty} \left(\mathbb{E}[\widehat{\theta}_j - \theta_j] \right)^2 + \sum_{j=1}^{\infty} \text{Var}(\widehat{\theta}_j) \right] \\ &= \frac{1}{\epsilon} \left[\sum_{j=1}^{\infty} \theta_j^2 \frac{1}{(1 + n \tau_j^2)^2} + \sum_{j=1}^{\infty} \frac{\tau_j^2}{1 + n \tau_j^2} \right] && \text{Theorem 6} \\ &\rightarrow 0, \end{aligned}$$

as $n \rightarrow \infty$, since $\sum_j \tau_j^2 < \infty$ and $\sum_j \theta_j^2 < \infty$.

(c) See [Shen and Wasserman \(2001\)](#).

Theorem 8 Let $X_1, \dots, X_n \sim N(\mu, \sigma^2)$ where σ^2 is known. Let $\mu \sim N(a, b^2)$. Then,

$$\mathbb{E}[\mu | X_1, \dots, X_n] = \frac{a\sigma^2 + n\bar{X}b^2}{\sigma^2 + nb^2}.$$

Proof.

$$\begin{aligned} f_{X^n}(x^n | \mu) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{\sigma^2}(x_i - \mu)^2\right\} = (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2\right\} \\ &= (2\pi\sigma^2)^{-n/2} \exp\left\{\frac{-1}{2\sigma^2} [(n-1)s^2 + n(\mu - \bar{X})^2]\right\} = (2\pi\sigma^2)^{-n/2} \exp\left\{\frac{-(n-1)s^2}{2\sigma^2}\right\} \exp\left\{\frac{-n(\mu - \bar{X})^2}{2\sigma^2}\right\} \\ &\propto \exp\left\{\frac{-n(\mu - \bar{X})^2}{2\sigma^2}\right\}. \\ \pi(\mu) &= \frac{1}{\sqrt{2\pi b^2}} \exp\left\{-\frac{1}{2b^2}(\mu - a)^2\right\} \propto \exp\left\{-\frac{1}{2b^2}(\mu - a)^2\right\}. \end{aligned}$$

Hence,

$$\begin{aligned} \pi(\mu | X^n) &\propto f_{X^n}(x^n | \mu) \pi(\mu) \propto \exp\left\{\frac{-n(\mu - \bar{X})^2}{2\sigma^2} - \frac{1}{2b^2}(\mu - a)^2\right\} \\ &= \exp\left\{\frac{-n\mu^2 + 2n\mu\bar{X} - n\bar{X}^2}{2\sigma^2} + \frac{-\mu^2 + 2\mu a - a^2}{2b^2}\right\} \\ &= \exp\left\{\mu^2\left(\frac{-1}{2b^2} - \frac{n}{2\sigma^2}\right) - 2\mu\left(-\frac{a}{2b^2} - \frac{n\bar{X}}{2\sigma^2}\right) - \left(\frac{a^2}{2b^2} + \frac{n\bar{X}^2}{2\sigma^2}\right)\right\} \end{aligned}$$

For simplicity, let

$$U = \left(\frac{-1}{2b^2} - \frac{n}{2\sigma^2}\right) \quad \text{and} \quad V = \left(-\frac{a}{2b^2} - \frac{n\bar{X}}{2\sigma^2}\right).$$

Then

$$\begin{aligned} \pi(\mu | X^n) &\propto \exp\{U\mu^2 - 2V\mu\} = \exp\left\{U\left(\mu^2 - 2\mu\frac{V}{U} + \frac{V^2}{U^2}\right) - \frac{V^2}{U}\right\} \\ &\propto \exp\left\{U(\mu - \frac{V}{U})^2\right\} = \exp\left\{\frac{-1}{2(1/\sqrt{-2U})^2}(\mu - \frac{V}{U})^2\right\} \propto N\left(\frac{V}{U}, \frac{-1}{2U}\right). \end{aligned}$$

Therefore the mean of the posterior is,

$$\widehat{\mu} = \mathbb{E}[\mu | X^n] = \frac{V}{U} = \frac{-a\sigma^2 - nb^2\bar{X}}{-\sigma^2 - nb^2} = \boxed{\frac{a\sigma^2 + n\bar{X}b^2}{\sigma^2 + nb^2}}.$$