Statistical Methods for Machine Learning

Larry Wasserman

Carnegie Mellon University

Lecture Notes

https://www.stat.cmu.edu/~larry/=sml/

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. Gyorfi, 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 compuational 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 are 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) and 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 $Z_1, Z_2, \ldots, Z_n$ are iid with mean μ and $\mathbb{P}(a \leq Z_i \leq b) = 1$, then for any $\epsilon > 0$*

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

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

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

$$\mathsf{VC}(\mathcal{A}) = \sup\Big\{n : \ s_n(\mathcal{A}) = 2^n\Big\}. \tag{2}$$

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 $n \geq d$,*

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

Given data $Z_1, \ldots, Z_n \sim P$. 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 $t > \sqrt{2/n}$,*

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

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

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

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

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

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 $Y_1, \ldots, Y_n$ are iid with mean μ, $\mathrm{Var}(Y_i) \leq \sigma^2$ and $|Y_i| \leq M$. Then

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

It follows that

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

for small enough ϵ and c.

3 Probability

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

2. $X_n \rightsquigarrow Z$ means that $\mathbb{P}(X_n \leq z) \to \mathbb{P}(Z \leq z)$ 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 $\epsilon > 0$

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

5. Law of large numbers: $X_1, \ldots, X_n \sim P$ 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: $X_1, \ldots, X_n \sim P$ 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 $\widehat{\theta}$ be an estimator of θ. Then

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

where bias $= \mathbb{E}[\widehat{\theta}] - \theta$ and Var $= \text{Var}(\widehat{\theta})$. 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 $X_1, \ldots, X_n \sim P$ where $P \in \mathcal{P}$. Let $\theta = T(P)$ be some quantity of interest, Then $C_n = C(X_1, \ldots, X_n)$ 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 $X_1, \ldots, X_n \sim p_\theta$. MLE $\widehat{\theta}_n$ (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
$$\frac{\widehat{\theta}_n - \theta}{s_n} \rightsquigarrow N(0,1)$$
where $s_n = \sqrt{\frac{1}{nI(\widehat{\theta})}}$.
7. Asymptotic $1-\alpha$ confidence interval $C_n = \widehat{\theta}_n \pm z_{\alpha/2}\ s_n$. Then
$$\mathbb{P}(\theta \in C_n) \to 1-\alpha.$$

5 Minimaxity

Let $\mathcal{P}$ be a set of distributions. Let θ be a parameter and let $L(\widehat{\theta}, \theta)$ be a loss function. The **minimax risk** is
$$R_n = \inf_{\widehat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[L(\widehat{\theta}, \theta)].$$
If $\sup_{P\in\mathcal{P}} \mathbb{E}_P[L(\widehat{\theta},\theta)] = R_n$ then $\widehat{\theta}$ is a minimax estimator.

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

As another example, if $X_1, \ldots, X_n \sim p$ where $X_i \in \mathbb{R}^d$, $L(\widehat{p}, p) = \int (\widehat{p} - p)^2$ 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 $X = (X(1), \ldots, X(d))$.
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
$$(X_1, Y_1), \ldots, (X_n, Y_n).$$
Given new X predict Y.
5. Minimize training error
$$\widehat{R}(\beta) = \frac{1}{n}\sum_i (Y_i - \beta^T X_i)^2.$$
Solution: least squares:
$$\widehat{\beta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T Y$$
where $\mathbb{X}(i,j) = X_i(j)$.
6. Fitted values $\widehat{Y} = \mathbb{X}\widehat{\beta} = HY$ 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 $\widehat{Y} = \widehat{m}(X)$ where $\widehat{m}(x) = x^T\widehat{\beta}$. Then
$$R = \mathbb{E}(\widehat{Y} - Y)^2 = \sigma^2 + \int b^2(x)p(x)dx + \int v(x)p(x)dx$$
where $b(x) = \mathbb{E}[\widehat{m}(x)] - m(x)$, $v(x) = \mathrm{Var}(\widehat{m}(x))$ and $\sigma^2 = \mathrm{Var}(\epsilon)$.

7 Classification

1. $X \in \mathbb{R}^d$ and $Y \in \{0, 1\}$.
2. Classifier $h : \mathbb{R}^d \to \{0, 1\}$.
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
$$h(x) = \mathrm{sign}(\psi(x))$$
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 $x = (1, x(2), \ldots, x(d))$.) Given data $(X_1, Y_1), \ldots, (X_n, Y_n)$ there are several ways to estimate a linear classifier:
(a) Empirical risk minimization (ERM): Choose $\widehat{\beta}$ 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 $Y_i \sim$ Benoulii$(p(X_i, \beta))$. 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 $\widehat{\beta}$ easily. It is easy to check that the classifier $I(p_{x,\widehat{\beta}} > 1/2)$ 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 $X_1, \ldots, X_n$ 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 $\widehat{p}$. The estimator will depend on a smoothing parameter h and choosing h carefully is crucial. To emphasize the dependence on h we sometimes write $\widehat{p}_h$.

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

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

where $\widehat{p}(y|x) = \widehat{p}(y, x)/\widehat{p}(x)$. 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 $X_1, \ldots, X_n$.

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

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

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.

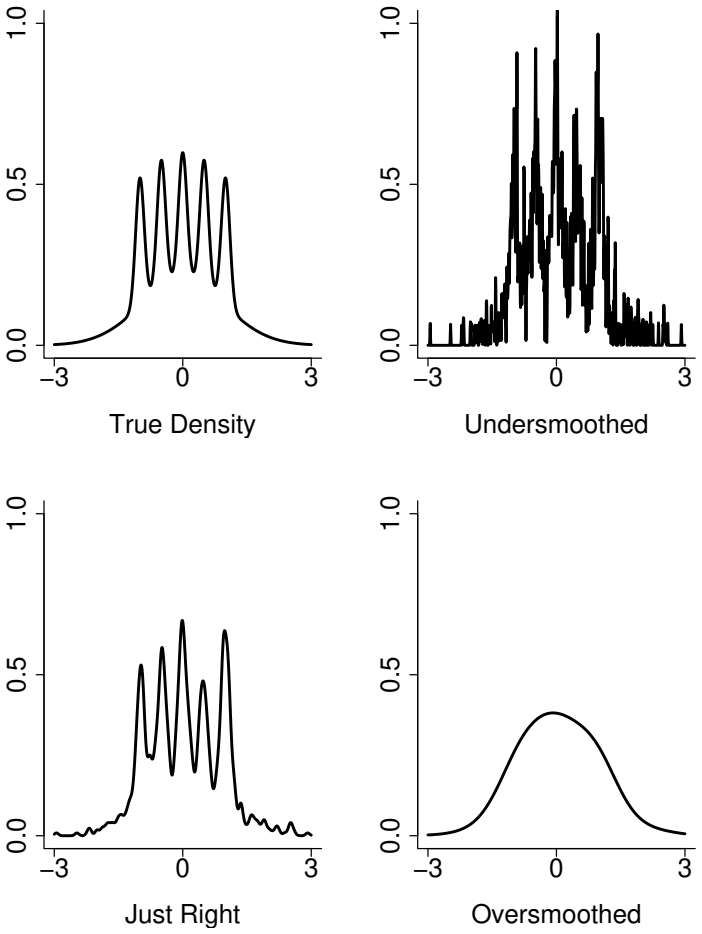


Figure 1: The Bart Simpson density from Example 1. Top left: true density. The other plots are kernel estimators based on $n = 1{,}000$ draws. Bottom left: bandwidth $h = 0.05$ chosen by leave-one-out cross-validation. Top right: bandwidth $h/10$. Bottom right: bandwidth $10h$.

2 Loss Functions

The most commonly used loss function is the L_2 loss

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

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

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

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

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

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

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 $\widetilde{p}$ and $\widetilde{q}$ be the corresponding densities for Y. Then

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

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. **H**

3 Histograms

Perhaps the simplest density estimators are histograms. For convenience, assume that the data $X_1, \ldots, X_n$ 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 $B_1, \ldots, B_N$. The histogram density estimator is

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

where

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

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

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

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

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

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,

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

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

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

We conclude that the L_2 risk is bounded by

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

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

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

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

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*

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

3.1 Concentration Analysis For Histograms

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

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

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

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

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

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

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

$$\mathbb{P}(|\widehat{\theta}_j - \theta_j| > h^d \epsilon) \leq 2h^{-d}\exp\left(-cn\epsilon^2 h^d\right) \equiv \pi_n.$$

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

$$\|\widehat{p}_h - p\|_\infty \leq \|\widehat{p}_h - p_h\|_\infty + \|p_h - p\|_\infty \leq \epsilon + L\sqrt{d}h. \tag{7}$$

Now set

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

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

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

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

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

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 $\sigma_K^2 \equiv \int x^2K(x)dx > 0$. *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:**	$K(x) = \frac{70}{81}(1-\lvert x\rvert^3)^3 I(x)$

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 $K(\|x\|)$.

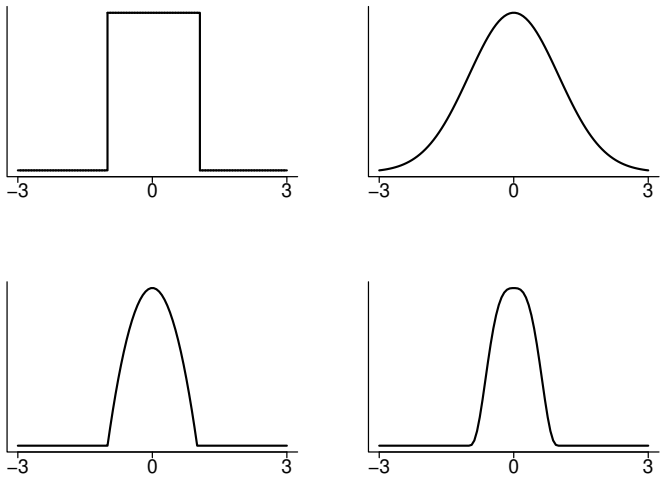


Figure 2: Examples of smoothing kernels: boxcar (top left), Gaussian (top right), Epanechnikov (bottom left), and tricube (bottom right).

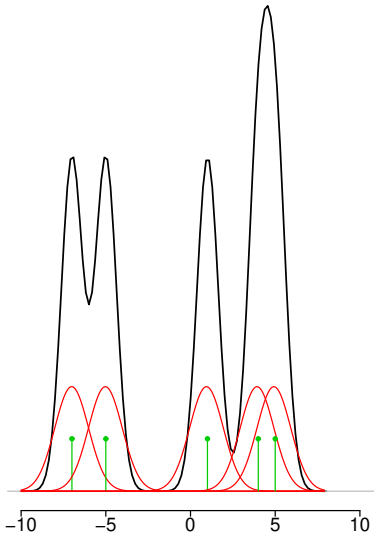


Figure 3: A kernel density estimator $\widehat{p}$. At each point x, $\widehat{p}(x)$ is the average of the kernels centered over the data points X_i. The data points are indicated by short vertical bars. The kernels are not drawn to scale.

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

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

More generally, we define

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

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 $H = h^2 I$ and we get back the previous formula.

Sometimes we write the estimator as $\widehat{p}_h$ 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 $s = (s_1, \ldots, s_d)$, define $|s| = s_1 + \cdots + s_d$, $s! = s_1! \cdots s_d!$, $x^s = x_1^{s_1} \cdots x_d^{s_d}$ and

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

Let β be a positive integer. Define the Hölder class

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

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:

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

where **H**

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

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

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

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

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). □

Next we bound the variance.

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

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

for some $c > 0$.

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

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

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

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*

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

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*

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

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 $\widehat{p}(x)$ concentrates around $p(x)$. First, recall Bernstein's inequality: Suppose that $Y_1, \ldots, Y_n$ are iid with mean μ, $\mathrm{Var}(Y_i) \leq \sigma^2$ and $|Y_i| \leq M$. Then

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

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

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

Hence, for any $\delta > 0$,

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

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

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

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,

$$|\widehat{p}(x) - p(x)| \leq |\widehat{p}(x) - p_h(x)| + |p_h(x) - p(x)| \tag{21}$$

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

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

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

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

The result follows from (21). □

4.4 Concentration in L_∞

Theorem 7 shows that, for each x, $\widehat{p}(x)$ 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

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

We can write

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

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

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

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

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

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,*

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

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

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

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*

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

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

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

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 $s_n(x) = \sqrt{\text{Var}(\widehat{p}_h(x))}$. The CLT implies that

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

for some $\tau(x)$. This is true even if $h = h_n$ is decreasing. Specifically, suppose that $h_n \to 0$ and $nh_n \to \infty$. 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

$$\lim_{n\to\infty} \sum_{i=1}^n \mathbb{E}[L_{n,i}|^{2+\delta} = 0$$

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

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

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

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

This means that the usual confidence interval $\widehat{p}_h(x) \pm z_{\alpha/2} s(x)$ 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 $\widehat{p}_h(x) \pm z_{\alpha/2} s(x)$ 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 $X_1, \ldots, X_n$. The idea is to estimate the distribution

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

with the bootstrap estimator

$$\widehat{F}_n(t) = \mathbb{P}\Big(\sqrt{nh^d}||\widehat{p}^*_h(x) - \widehat{p}_h(x)||_\infty \leq t \;\Big|\; X_1, \ldots, X_n\Big)$$

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

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

Here is the algorithm.

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

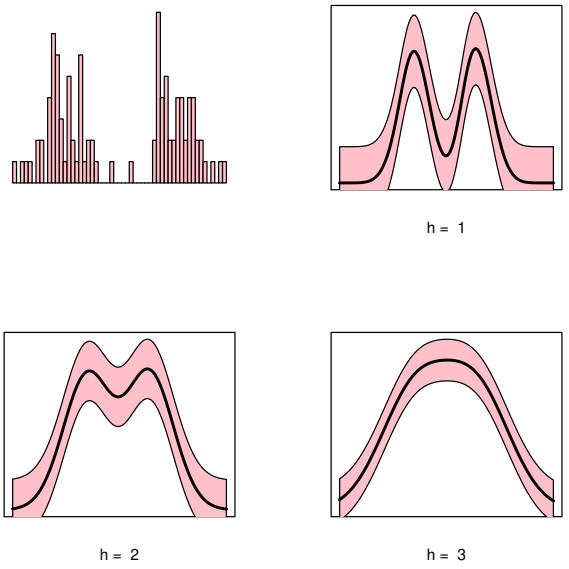


Figure 4: 95 percent bootstrap confidence bands using various bandwidths.

2. Draw $X_1^*, \ldots, X_n^* \sim P_n$. This is called a bootstrap sample.
3. Compute the density estimator $\widehat{p}_h^*$ based on the bootstrap sample.
4. Compute $R = \sup_x \sqrt{nh^d}||\widehat{p}_h^* - \widehat{p}_h||_\infty$.
5. Repeat steps 2-4 B times. This gives $R_1, \ldots, R_B$.
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

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

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

$$\liminf_{n\to\infty} \mathbb{P}\big(\ell_n(x) \le p_h(x) \le u(x) \quad \text{for all } x\big) \ge 1-\alpha.$$

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 $\widehat{p}_h$. Note that,

$$\mathbb{E}[\widehat{p}_h(x)] = p(x) + C(x)h^2 + o(h^2)$$
$$\mathbb{E}[\widehat{p}_{2h}(x)] = p(x) + C(x)4h^2 + o(h^2)$$

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.

H

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,

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

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

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:

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

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,*

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

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 $V = 10$. Fir 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 $X = (X_1, \ldots, X_{2n})$ into two sets of size n. Denote these by $Y = (Y_1, \ldots, Y_n)$ and $Z = (Z_1, \ldots, Z_n)$.[1] Let $\mathcal{H} = \{h_1, \ldots, h_N\}$ be a finite grid of bandwidths. Let

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

Thus we have a set $\mathcal{P} = \{\widehat{p}_1, \ldots, \widehat{p}\}$ of density estimators.

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

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

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

$$X = (X_1, \ldots, X_{2n}) \overset{\text{split}}{\Longrightarrow} \begin{array}{l} Y \to \{\widehat{p}_1, \ldots, \widehat{p}_N\} = \mathcal{P} \\ \\ Z \to \{\widehat{L}_1, \ldots, \widehat{L}_N\} \end{array}$$

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

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

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

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

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

and

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

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

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

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

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,

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

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

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

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

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 $p_h(x) = \mathbb{E}(\widehat{p}_h(x))$. Then

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

so that the mean of $\widehat{p}_h$ 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 convolution[2] 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 $\mathbb{P}(\|\widehat{p}_h - p_h\|_\infty > \epsilon) \leq Ce^{-nc\epsilon^2}$. Hence,*

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

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 $(X_1, \ldots, X_d)$ as if they are independent. In that case

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

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

[2]If $X \sim P$ 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

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

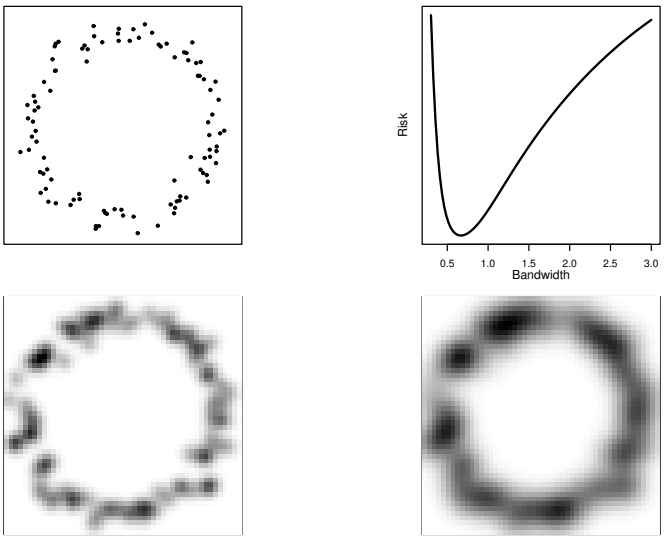


Figure 5: Synthetic two-dimensional data set. Top left: data. Top right: cross-validation function. Bottom left: kernel estimator based on the bandwidth that minimizes the cross-validation score. Bottom right: kernel estimator based on the twice the bandwidth that minimizes the cross-validation score.

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

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

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

$$\widehat{p}_A(Y_1), \ldots, \widehat{p}_A(Y_{n+1}).$$

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

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

A p-value for the test is

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

The prediction set is

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

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 $Z_i = \widehat{p}(Y_i)$. Order these observations

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

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

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

Define

$$C_n^+ = \Big\{ y : \ \widehat{p}(y) \geq t \Big\}.$$

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, asympotic 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

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

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 $\phi_1, \phi_2, \ldots$ be an orthonormal basis for

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

Thus

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

An example is the cosine basis:

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

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 $\widehat{p}(x) = \sum_{j=1}^{k} \widehat{\beta}_j \phi_j(x)$ where

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

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

It can be shown that

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

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

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

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 $\int_A \widehat{p}_h(x)dx$ close to $P(A)$ for all $A \in \mathcal{A}$. That is, we would like to minimize

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

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 $\mathcal{P} = \{\widehat{p}_1, \ldots, \widehat{p}_N\}$ 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 $\widehat{p} = \text{argmin}_{g \in \mathcal{P}} \Delta_n(g)$.

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

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

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 $p_* = \text{argmin}_{g \in \mathcal{P}} \Delta(g)$. Then,

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

□

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 $\mathcal{P} = \{p_1, \ldots, p_N\}$ be a set of densities and define the Yatracos class of sets $\mathcal{A} = \Big\{ A(i,j) : \ i \neq j \Big\}$ where $A(i,j) = \{x : p_i(x) > p_j(x)\}$. Let

$$\widehat{p} = \text{argmin}_{g \in \mathcal{G}} \Delta(g)$$

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 $Z_1, \ldots, Z_n \sim p$.

Theorem 18 *The estimator $\widehat{p}$ satisfies*

$$\int |\widehat{p} - p| \leq 3 \min_j \int |p_j - p| + 4\Delta \tag{34}$$

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

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 $\widehat{p} = p_i$ 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,

$$\int |\widehat{p} - p| \leq \int |p_s - p| + \int |p_s - p_i|. \tag{35}$$

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

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

The result follows from (35). □

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

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

Let

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

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

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

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

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

$$\mathbb{E}\int |\widehat{p} - p| \leq c_1 \inf_h \mathbb{E}\int |\widehat{p}_h - p| + c_2\sqrt{\frac{\log n}{n}}. \tag{36}$$

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 $X_1, \ldots, X_n \sim p$ and $Y_1, \ldots, Y_m \sim q$ we can test $H_0 : p = q$ using $\int(\widehat{p} - \widehat{q})^2$ 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 $\widehat{p}(x) - \widehat{q}(x)$. 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

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

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

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

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* $X_i : [0, T_i] \to \mathbb{R}^2$ *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*

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

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*

[4]Thanks to Susan Buchman for providing the data.

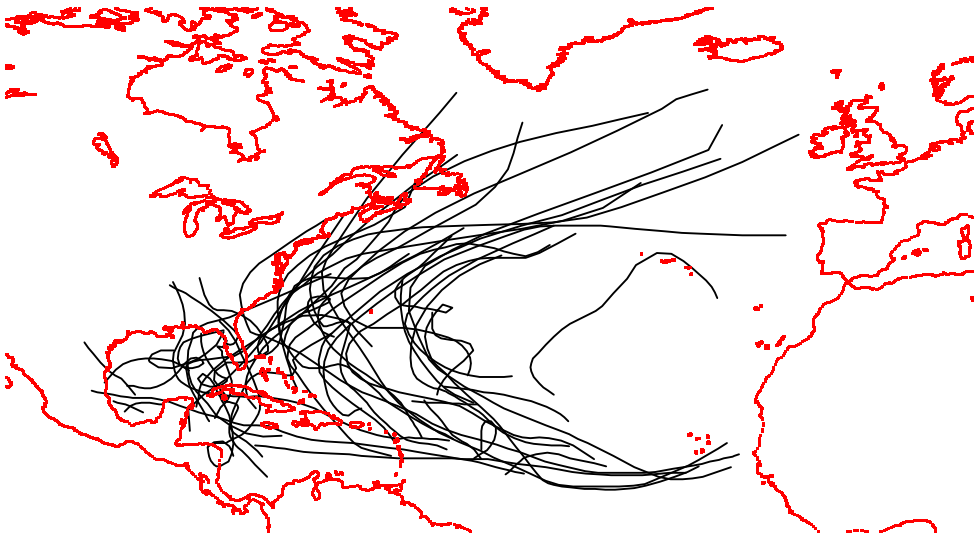


Figure 6: Paths of 40 tropical cyclones in the North Atlantic.

distance between two sets A is B is

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

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

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

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

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 $d = 1$

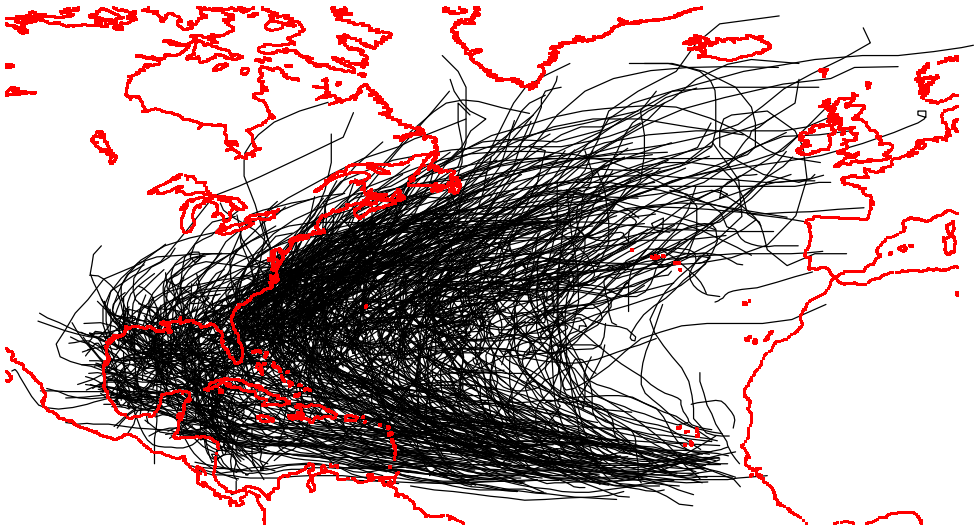


Figure 7: Paths of 608 tropical cyclones in the North Atlantic.

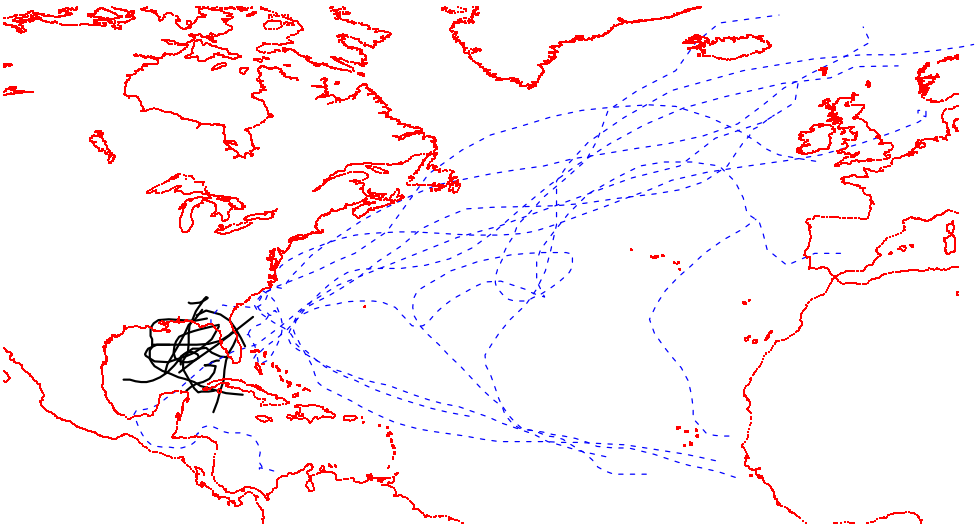


Figure 8: 10 highest density paths (black) and 10 lowest density paths (blue).

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.

[5] Recall 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 $\widehat{m}$ of m_0, from i.i.d. samples $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$. Given a new X, our prediction of Y is $\widehat{m}(X)$. We often call X the input, predictor, feature, etc., and Y the output, outcome, response, etc.

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

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

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

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

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

For arbitrary X_i, $i = 1, \ldots, n$, this is really just the same as starting with the random input model, and conditioning on the particular values of X_i, $i = 1, \ldots, n$. (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

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

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, $i = 1, \ldots, n$, acting on functions $m : \mathbb{R}^d \to \mathbb{R}$, 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 $m : \mathbb{R}^d \to \mathbb{R}$, 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 $\widehat{m}$ of m_0, which can be measured in either norm:

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

In either case, this is a random quantity (since $\widehat{m}$ 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

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

where $b_n(x) = \mathbb{E}[\widehat{m}(x)] - m(x)$ is the bias, $v(x) = \mathrm{Var}(\widehat{m}(x))$ 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 $\widehat{m}(x) = \sum_{j=1}^p \widehat{\beta}_j g_j(x)$. A common question: the coefficients on the spline basis functions $\beta_1, \ldots, \beta_p$ 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 $g_1, \ldots, g_p$.

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

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

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

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

(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

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

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

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 $s = (s_1, \ldots, s_d)$, define $|s| = s_1 + \cdots + s_d$, $s! = s_1! \cdots s_d!$, $x^s = x_1^{s_1} \cdots x_d^{s_d}$ and

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

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

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

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:

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

where

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

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) $g : \mathbb{R} \to \mathbb{R}$ 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

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

where $\mathcal{N}_k(x)$ contains the indices of the k closest points of $X_1, \ldots, X_n$ 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 $\widehat{m}$, 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 $\widehat{m}$ essentially always looks jagged, especially for small or moderate k. Why is this? It helps to write

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

where the weights $w_i(x)$, $i = 1, \ldots, n$ 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 $\widehat{m}(x)$.

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

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

can simply be expressed as $\widehat{\mu} = SY$. (To be clear, this means that for fixed inputs $X_1, \ldots, X_n$, the vector of fitted values $\widehat{\mu}$ is a linear function of Y; it does not mean that $\widehat{m}(x)$ 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 $\mathbb{E}\|\widehat{m} - m_0\|_2^2 \to 0$ as $n \to \infty$, with no assumptions other than $\mathbb{E}(Y^2) \leq \infty$, provided that we take $k = k_n$ such that $k_n \to \infty$ and $k_n/n \to 0$; 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

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

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

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

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

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

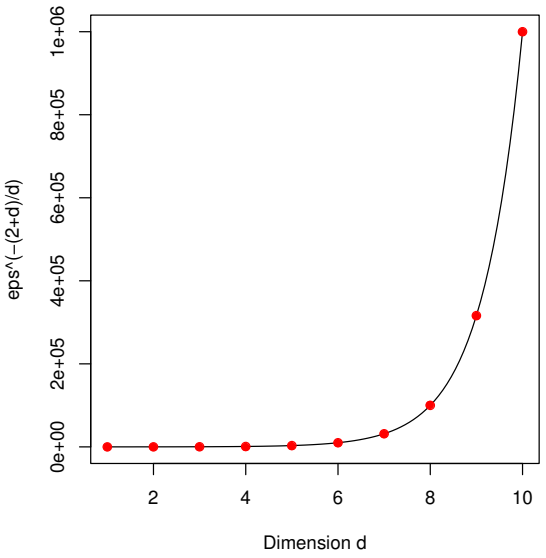


Figure 1: *The curse of dimensionality, with* $\epsilon = 0.1$

constant $C > 0$. (Think of a having input points X_i, $i = 1, \ldots, n$ 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

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

where the infimum above is over all estimators $\widehat{m}$. 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 $K : \mathbb{R} \to \mathbb{R}$, 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

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

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 ($\mathbb{E}\|\widehat{m} - m_0\|_2^2 \to 0$ as $n \to \infty$, 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 $h_n \to 0$ and $nh_n^d \to \infty$ as $n \to \infty$. See Chapter 5.2 of Gyorfi et al. (2002). We can say more.

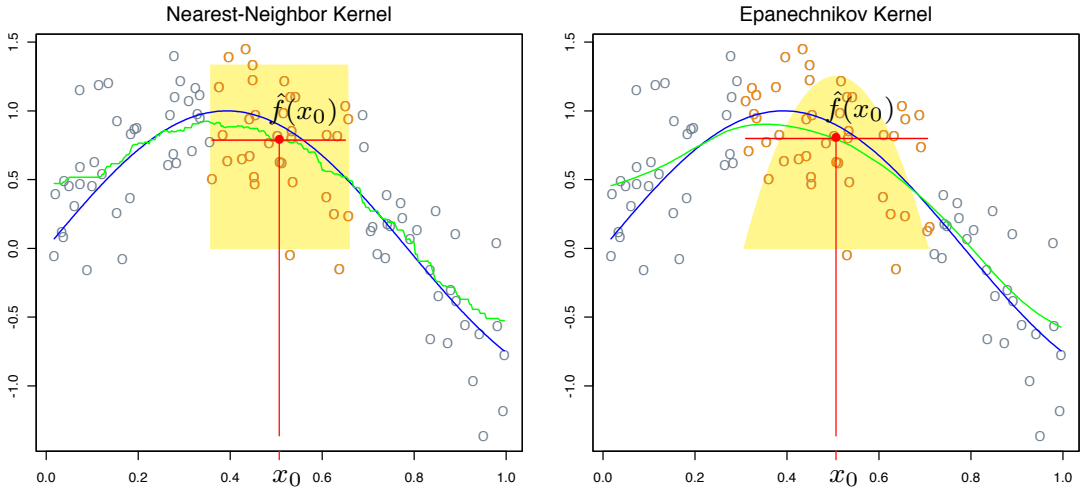


Figure 2: *Comparing k-nearest-neighbor and Epanechnikov kernels, when* $d = 1$. *From Chapter 6 of Hastie et al. (2009)*

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,

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

where P_n is the empirical measure and $B(x,h) = \{u : \|x-u\| \leq h\}$. If the denominator is 0 we define $\widehat{m}(x) = 0$. 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 $\mathrm{Var}(Y|X=x) \leq \sigma^2 < \infty$ for all x. Then

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

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

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

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

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

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 $\widehat{m}(x) = \widehat{\theta}_x$, where $\widehat{\theta}_x$ 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 $\widehat{m}(x) = \widehat{\alpha}_x + \widehat{\beta}_x^T x$, where $\widehat{\alpha}_x, \widehat{\beta}_x$ 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*.

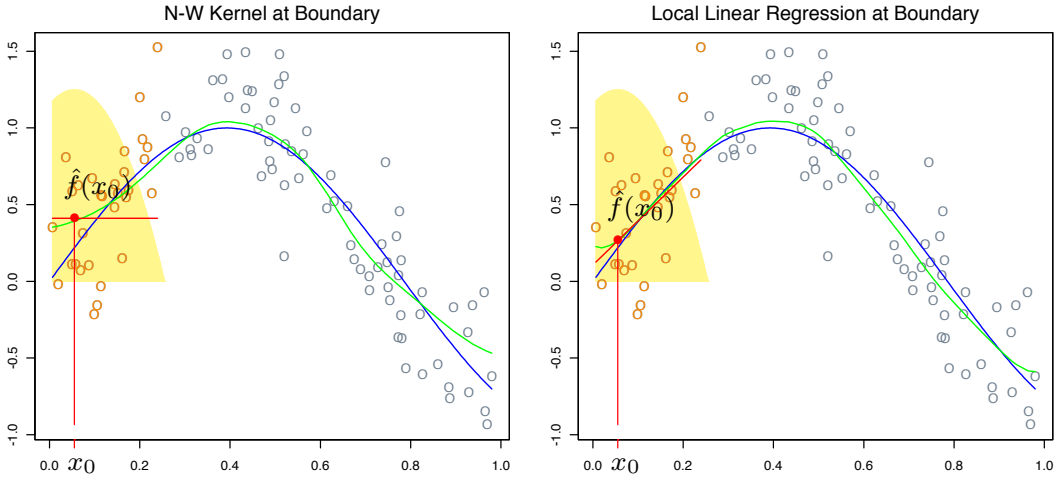


Figure 3: *Comparing (Nadaraya-Watson) kernel smoothing to local linear regression; the former is biased at the boundary, the latter is unbiased (to first-order). From Chapter 6 of Hastie et al. (2009)*

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

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

where $b(x) = (1, x) \in \mathbb{R}^{d+1}$, $B \in \mathbb{R}^{n \times (d+1)}$ with ith row $b(X_i)$, and $\Omega \in \mathbb{R}^{n \times n}$ is diagonal with ith diagonal element $K(\|x - X_i\|_2 / h)$. We can write more concisely as $\widehat{m}(x) = w(x)^T Y$, 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 $\widehat{\mu} = (\widehat{m}(x_1), \ldots, \widehat{m}(x_n))$ can be expressed as

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

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,

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

Using a Taylor expansion of m_0 about x,

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

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 $\widehat{m}$,

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

and so $\mathbb{E}[\widehat{m}(x)] = m_0(x) + R$, which means that $\widehat{m}$ 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 $\widehat{m}$ is

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

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 $\frac{R_n}{r_n} \to 1$.

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 $\widehat{m}(x) = \widehat{\beta}_{x,0} + \sum_{j=1}^k \widehat{\beta}_{x,j} x^j$, where $\widehat{\beta}_{x,0}, \ldots, \widehat{\beta}_{x,k}$ 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 $\beta_0, \beta_1, \ldots, \beta_k \in \mathbb{R}$. This is called ($k$th-order) *local polynomial regression*

Again we can express

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

where $b(x) = (1, x, \ldots, x^k)$, B is an $n \times (k+1)$ matrix with ith row $b(X_i) = (1, X_i, \ldots, X_i^k)$, 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 $\widehat{m}$ 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

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

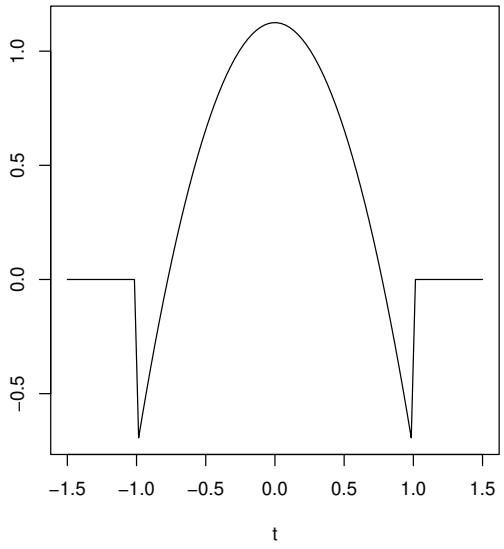


Figure 4: *A higher-order kernel function: specifically, a kernel of order 4*

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 kth 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

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

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

An example of a 4th-order kernel is $K(t) = \frac{3}{8}(3 - 5t^2)1\{|t| \leq 1\}$, 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

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

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 $\eta_1, \ldots, \eta_n$ for the set of kth-order natural splines with knots over $x_1, \ldots, x_n$, and reparametrize the problem as

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

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

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

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

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

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

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

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

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

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

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 $f : \mathbb{R} \to \mathbb{R}$ such that

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

(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 $\widehat{m}$ of polynomial order $k = 2m - 1$ with tuning parameter $\lambda \asymp n^{1/(2m+1)} \asymp n^{1/(k+2)}$ satisfies

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

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

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

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

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 $f_n \to f$ 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 $||f_n - f|| = 1/\sqrt{n} \to 0$. 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 unsmooth 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 $(X_1, Y_1), \ldots, (X_n, Y_n)$ and we want to estimate $m(x) = \mathbb{E}(Y|X = x)$. The approach we used earlier was based on **smoothing kernels**:

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

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: $x^T Ax \geq 0$ 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 $\alpha = (\alpha_1, \ldots, \alpha_k)^T$ 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:

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

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 $f_n \to f$. Then

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

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

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

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

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

We can expand f either in terms of K or in terms of the basis $\psi_1, \psi_2, \ldots$:

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

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 $(X_1, Y_1), \ldots, (X_n, Y_n)$ and on $f(X_1), \ldots, f(X_n)$. Let $\widehat{f}$ minimize

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

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

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

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

5.9 RKHS Regression

Define $\widehat{m}$ to minimize

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

By the representer theorem, $\widehat{m}(x) = \sum_{i=1}^n \alpha_i K(x_i, x)$. 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

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

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

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

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 $\widehat{f} = \sum_j K(x_j, x)$ 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 $\widehat{m}(x) = \sum_i w_i(x) Y_i$ for some weights $w_i(x)$ that do not depend on the $Y_i's$. Hence, the fitted values $\widehat{\mu} = (\widehat{m}(X_1), \ldots, \widehat{m}(X_n))$ are of the form $\widehat{\mu} = SY$ for some matrix $S \in \mathbb{R}^{n \times n}$ depending on the inputs $X_1, \ldots, X_n$—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, $\widehat{\mu} = HY$ for some projection matrix H.

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

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

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

$$\mathrm{CV}(\widehat{m}) = \frac{1}{n} \sum_{i=1}^{n} \big(Y_i - \widehat{m}^{-i}(X_i)\big)^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{Y_i - \widehat{m}(X_i)}{1 - S_{ii}} \right)^2, \tag{17}$$

where $\widehat{m}^{-i}$ denotes the estimated regression function that was trained on all but the ith 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 $\widehat{m}^{-i}$, $i = 1, \ldots, n$.

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

$$\widehat{m}^{-i}(X_i) = \frac{1}{1 - S_{ii}} \big(\widehat{m}(X_i) - S_{ii} Y_i\big). \tag{18}$$

That is, to obtain the estimate at X_i under the function $\widehat{m}^{-i}$ fit on all but (X_i, Y_i), we take the sum of the linear weights (from our original fitted function $\widehat{m}$) across all but the ith point, $\widehat{m}(X_i) - S_{ii} Y_i = \sum_{i \neq j} S_{ij} Y_j$, 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

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

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 $\mathrm{tr}(S)/n$,

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

where ν is the effective degrees of freedom and $\widehat{R}$ is the training error. This can be of computational advantage in some cases where $\mathrm{tr}(S)$ is easier to compute that 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

$$m(x) = m(x(1), \ldots, x(d)) \tag{19}$$

we restrict our attention to functions of the form

$$m(x) = m_1(x(1)) + \cdots + m_d(x(d)). \tag{20}$$

As each function m_j, $j = 1, \ldots, d$ 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* $\overline{m}^{\text{add}}$ well. Assuming each component function $\overline{m}^{\text{add}}_{0,j}$, $j = 1, \ldots, d$ lies in the Holder class $H_1(\alpha, L)$, for constant $L > 0$, and we can use an additive model, with each component $\widehat{m}_j$, $j = 1, \ldots, d$ estimated using an appropriate kth degree spline, to give

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

Hence each component of the best additive approximation $\overline{f}^{\text{add}}$ 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, $j = 1, \ldots, d$ individually (like a block coordinate descent algorithm). Denote the result of running our chosen univariate smoother to regress $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$ over the input points $Z = (Z_1, \ldots, Z_n) \in \mathbb{R}^n$ as

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

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 $\widehat{m}_1, \ldots, \widehat{m}_d$ (say, to all to zero) and cycle over the following steps for $j = 1, \ldots, d, 1, \ldots, d, \ldots$:

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

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

$$\widehat{m}(x) = \overline{Y} + \widehat{m}_1(x(1)) + \cdots + \widehat{m}(x(d))$$

where $\overline{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$. 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,

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

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 $\text{Smooth}_j(\cdot, \cdot)$, $j = 1, \ldots, d$ 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 $f_{0,j} \in S_1(m, C)$, $j = 1, \ldots, d$, for a constant $C > 0$, a somewhat straightforward argument building on univariate minimax theory gives us the lower bound

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

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 $S \subseteq \{1, \ldots, d\}$ 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

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

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

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

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 $\widehat{m}(x)$. Suppose that $\widehat{m}(x) = \sum_i w_i(x) Y_i$. The conditional variance is $\sum_i w_i^2(x)\sigma^2(x)$ which can

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

A better idea is to bootstrap the quantity

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

to get a bootstrap quantile t_n. Then

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

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, $\phi_1, \ldots, \phi_n$, evaluated over n equally spaced inputs over $[0, 1]$:

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

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

$$W_{ij} = \phi_j(X_i), \quad i, j = 1, \ldots, n$$

The goal, given outputs $y = (y_1, \ldots, y_n)$ 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):

$$\widetilde{\theta} = W^T y,$$

we threshold the coefficients θ (the threshold function T_λ to be defined shortly):

$$\widehat{\theta} = T_\lambda(\widetilde{\theta}),$$

and then perform an inverse wavelet transform (multiply by W):

$$\widehat{\mu} = W\widehat{\theta}$$

The wavelet and inverse wavelet transforms (multiplication by W^T and W) each require $O(n)$ operations, and are practically extremely fast due do 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.,

$$[T_\lambda^{\text{hard}}(z)]_i = z_i \cdot 1\{|z_i| \geq \lambda\}, \quad i = 1, \ldots, n,$$

or soft-thresholding, i.e.,

$$[T_\lambda^{\text{soft}}(z)]_i = \big(z_i - \text{sign}(z_i)\lambda\big) \cdot 1\{|z_i| \geq \lambda\}, \quad i = 1, \ldots, n.$$

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 $\widehat{\mu} = Sy$ 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

$$\widehat{\theta} = \underset{\theta \in \mathbb{R}^n}{\text{argmin}} \ \|y - W\theta\|_2^2 + \lambda^2 \|\theta\|_0,$$

and then the wavelet smoothing fitted values are $\widehat{\mu} = W\widehat{\theta}$. Here $\|\theta\|_0 = \sum_{i=1}^n 1\{\theta_i \neq 0\}$, the number of nonzero components of θ, called the "ℓ_0 norm". For soft-thresholding, we solve

$$\widehat{\theta} = \underset{\theta \in \mathbb{R}^n}{\text{argmin}} \ \|y - W\theta\|_2^2 + 2\lambda \|\theta\|_1,$$

and then the wavelet smoothing fitted values are $\widehat{\mu} = W\widehat{\theta}$. 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 kth derivative satisfies

$$\text{TV}(f^{(k)}) = \sup_{0 = z_1 < z_2 < \ldots < z_N < z_{N+1} = 1} \sum_{j=1}^N |f^{(k)}(z_{i+1}) - f^{(k)}(z_i)| \ \leq \ C.$$

(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 $\widehat{m}^{\text{wav}}$, 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),

$$\mathbb{E}\|\widehat{m}^{\text{wav}} - m_0\|_2^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{and} \quad \inf_{\widehat{m}} \sup_{m_0 \in M(k,C)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-(2k+2)/(2k+3)}. \tag{21}$$

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

$$\inf_{\widehat{m} \text{ linear}} \sup_{m_0 \in M(k,C)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-(2k+1)/(2k+2)}. \tag{22}$$

(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 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

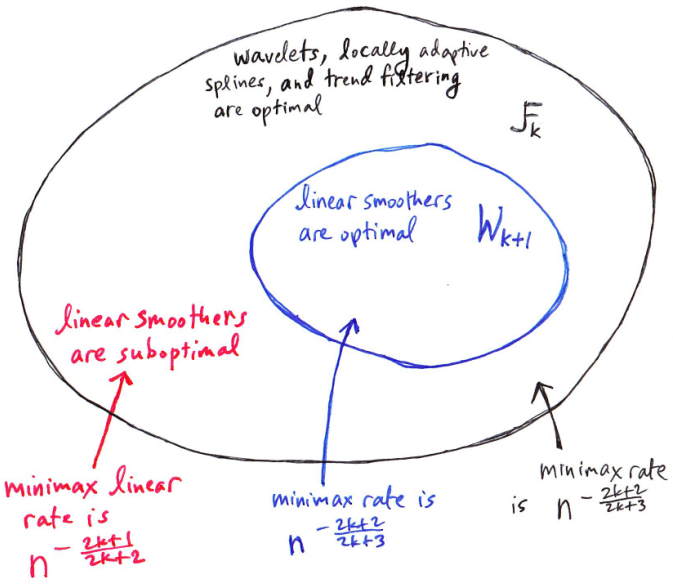


Figure 5: *A diagram of the minimax rates over $M(k, C)$ (denoted $\mathcal{F}_k$ in the picture) and $S_1(k+1, C)$ (denoted $\mathcal{W}_{k+1}$ in the picture)*

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 kth-order spline f is a piecewise polynomial function of degree k that is continuous and has continuous derivatives of orders $1, \ldots, k-1$, at its knot points. Specifically, there are $t_1 < \ldots < t_p$ such that f is a polynomial of degree k on each of the intervals

  $$(-\infty, t_1], [t_1, t_2], \ldots, [t_p, \infty)$$

  and $f^{(j)}$ is continuous at $t_1, \ldots, t_p$, for each $j = 0, 1, \ldots, k-1$
- 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 $t_1, \ldots, t_p$? The most natural way is to use the *truncated power basis*, $g_1, \ldots, g_{p+k+1}$, defined as

$$\begin{aligned} g_1(x) = 1, \; g_2(x) = x, \; \ldots \; g_{k+1}(x) = x^k, \\ g_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \ldots, p. \end{aligned} \tag{23}$$

  (Here x_+ denotes the positive part of x, i.e., $x_+ = \max\{x, 0\}$.) From this we can see that the space of kth-order splines with knots at $t_1, \ldots, t_p$ 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 $x_1, \ldots, x_n$ and responses $y_1, \ldots, y_n$, we consider fitting functions f that are kth-order splines with knots at some chosen locations $t_1, \ldots t_p$. This means expressing f as

$$f(x) = \sum_{j=1}^{p+k+1} \beta_j g_j(x),$$

  where $\beta_1, \ldots, \beta_{p+k+1}$ are coefficients and $g_1, \ldots, g_{p+k+1}$, are basis functions for order k splines over the knots $t_1, \ldots, t_p$ (e.g., the truncated power basis or B-spline basis)

- Letting $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$, and defining the basis matrix $G \in \mathbb{R}^{n \times (p+k+1)}$ by

$$G_{ij} = g_j(x_i), \quad i = 1, \ldots, n, \; j = 1, \ldots, p + k + 1,$$

  we can just use least squares to determine the optimal coefficients $\widehat{\beta} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_{p+k+1})$,

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^{p+k+1}}{\operatorname{argmin}} \; \|y - G\beta\|_2^2,$$

  which then leaves us with the fitted *regression spline* $\widehat{f}(x) = \sum_{j=1}^{p+k+1} \widehat{\beta}_j g_j(x)$

- Of course we know that $\widehat{\beta} = (G^T G)^{-1} G^T y$, so the fitted values $\widehat{\mu} = (\widehat{f}(x_1), \ldots, \widehat{f}(x_n))$ are

$$\widehat{\mu} = G(G^T G)^{-1} G^T y,$$

  and regression splines are linear smoothers

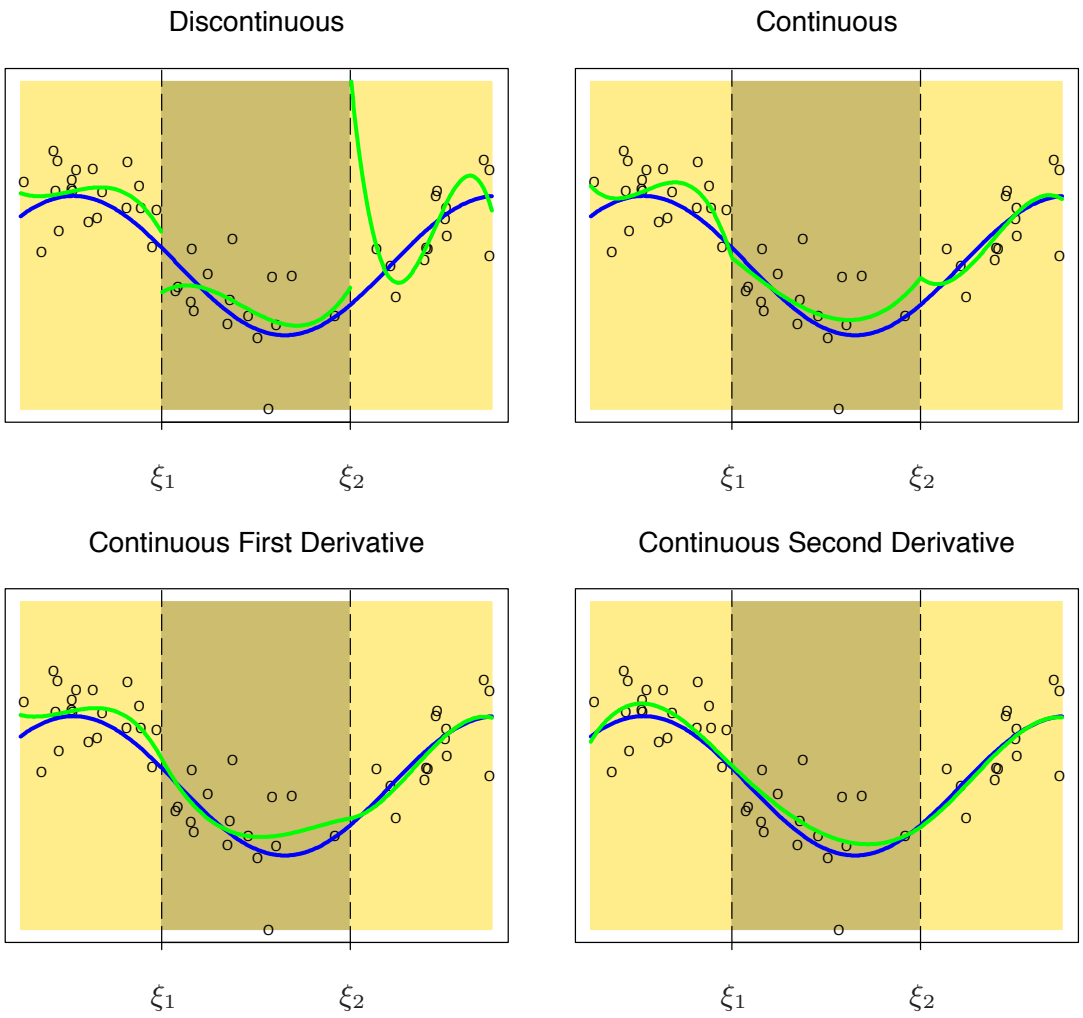


Figure 6: *Illustration of the effects of enforcing continuity at the knots, across various orders of the derivative, for a cubic piecewise polynomial. From Chapter 5 of Hastie et al. (2009)*

- This is a classic method, and can work well provided we choose good knots $t_1, \ldots, t_p$; 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 erractic 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 $t_1 < \ldots < t_p$, is a piecewise polynomial function f such that

  - f is a polynomial of degree k on each of $[t_1, t_2], \ldots, [t_{p-1}, t_p]$,
  - 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 $1, \ldots, k-1$ at $t_1, \ldots, t_p$.

  It is implicit here that natural splines are only defined for odd orders k

- What is the dimension of the span of kth order natural splines with knots at $t_1, \ldots, t_p$? 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:

$$\underbrace{(k+1)\cdot(p-1)}_{a} + \underbrace{\Big(\frac{(k-1)}{2}+1\Big)\cdot 2}_{b} - \underbrace{k\cdot p}_{c} = p.$$

  Above, a is the number of free parameters in the interior intervals $[t_1, t_2], \ldots, [t_{p-1}, t_p]$, 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 $t_1, \ldots, t_p$. 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 $x_1, \ldots, x_n$. 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 $x_1, \ldots, x_n$ lying in an interval $[0, 1]$, the *smoothing spline* estimate $\widehat{f}$, of a given odd integer order $k \geq 0$, is defined as

$$\widehat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \int_0^1 \big(f^{(m)}(x)\big)^2 \, dx, \quad \text{where } m = (k+1)/2. \tag{24}$$

  This is an infinite-dimensional optimization problem over all functions f for the which the criterion is finite. This criterion trades off the least squares error of f over the observed pairs (x_i, y_i), $i = 1, \ldots, n$, with a penalty term that is large when the mth 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

$$\widehat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \int_0^1 f''(x)^2 \, dx \tag{25}$$

- Remarkably, it so happens that the minimizer in the general smoothing spline problem (38) is unique, and is a natural kth-order spline with knots at the input points $x_1, \ldots, x_n$! 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 $\widetilde{f}$ is any twice differentiable function on $[0, 1]$, and $x_1, \ldots, x_n \in [0, 1]$, then there exists a natural cubic spline f with knots at $x_1, \ldots, x_n$ such that $f(x_i) = \widetilde{f}(x_i)$, $i = 1, \ldots, n$ and

$$\int_0^1 f''(x)^2 \, dx \leq \int_0^1 \widetilde{f}''(x)^2 \, dx.$$

  Note that this would in fact prove that we can restrict our attention in (25) to natural splines with knots at $x_1, \ldots, x_n$

  Proof: the natural spline basis with knots at $x_1, \ldots, x_n$ is n-dimensional, so given any n points $z_i = \widetilde{f}(x_i)$, $i = 1, \ldots, n$, we can always find a natural spline f with knots at $x_1, \ldots, x_n$ that satisfies $f(x_i) = z_i$, $i = 1, \ldots, n$. Now define

$$h(x) = \widetilde{f}(x) - f(x).$$

Consider

$$\begin{aligned}\int_0^1 f''(x)h''(x)\,dx &= f''(x)h'(x)\Big|_0^1 - \int_0^1 f'''(x)h'(x)\;dx \\ &= -\int_{x_1}^{x_n} f'''(x)h'(x)\;dx \\ &= -\sum_{j=1}^{n-1} f'''(x)h(x)\Big|_{x_j}^{x_{j+1}} + \int_{x_1}^{x_n} f^{(4)}(x)h'(x)\;dx \\ &= -\sum_{j=1}^{n-1} f'''(x_j^+)\big(h(x_{j+1}) - h(x_j)\big),\end{aligned}$$

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 $x \le x_1$ and $x \ge x_n$, 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}), $j = 1, \ldots, n-1$, 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 $j = 1, \ldots, n$, we have

$$\int_0^1 f''(x)h''(x)\,dx = 0.$$

From this, it follows that

$$\begin{aligned}\int_0^1 \widetilde{f}''(x)^2\,dx &= \int_0^1 \big(f''(x) + h''(x)\big)^2\,dx \\ &= \int_0^1 f''(x)^2\,dx + \int_0^1 h''(x)^2\,dx + 2\int_0^1 f''(x)h''(x)\,dx \\ &= \int_0^1 f''(x)^2\,dx + \int_0^1 h''(x)^2\,dx,\end{aligned}$$

and therefore

$$\int_0^1 f''(x)^2\,dx \le \int_0^1 \widetilde{f}''(x)^2\,dx, \tag{26}$$

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 $j = 1, \ldots, n$, this is equivalent to $h = 0$. In other words, the inequality (45) holds strictly except when $\widetilde{f} = f$, 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 $\eta_1, \ldots, \eta_n$ for the set of kth-order natural splines with knots over $x_1, \ldots, x_n$, and reparametrize the problem (38) as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^n \Big(y_i - \sum_{j=1}^n \beta_j \eta_j(x_i)\Big)^2 + \lambda \int_0^1 \Big(\sum_{j=1}^n \beta_j \eta_j^{(m)}(x)\Big)^2\,dx. \tag{27}$$

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

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

$$N_{ij} = \eta_j(x_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 \eta_i^{(m)}(x) \eta_j^{(m)}(x)\, dx \quad \text{for } i, j = 1, \ldots, n, \tag{28}$$

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

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - N\beta\|_2^2 + \lambda \beta \Omega \beta, \tag{29}$$

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

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

and therefore the fitted values $\widehat{\mu} = (\widehat{f}(x_1), \ldots, \widehat{f}(x_n))$ are

$$\widehat{\mu} = N(N^T N + \lambda \Omega)^{-1} N^T y. \tag{30}$$

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) is what is called Reinsch form,

$$\begin{aligned} \widehat{\mu} &= N(N^T N + \lambda \Omega)^{-1} N^T y \\ &= N\Big(N^T \big(I + \lambda (N^T)^{-1} \Omega N^{-1}\big) N\Big)^{-1} N^T y \\ &= (I + \lambda Q)^{-1} y, \end{aligned} \tag{31}$$

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 $D = \operatorname{diag}(d_1, \ldots, d_n)$

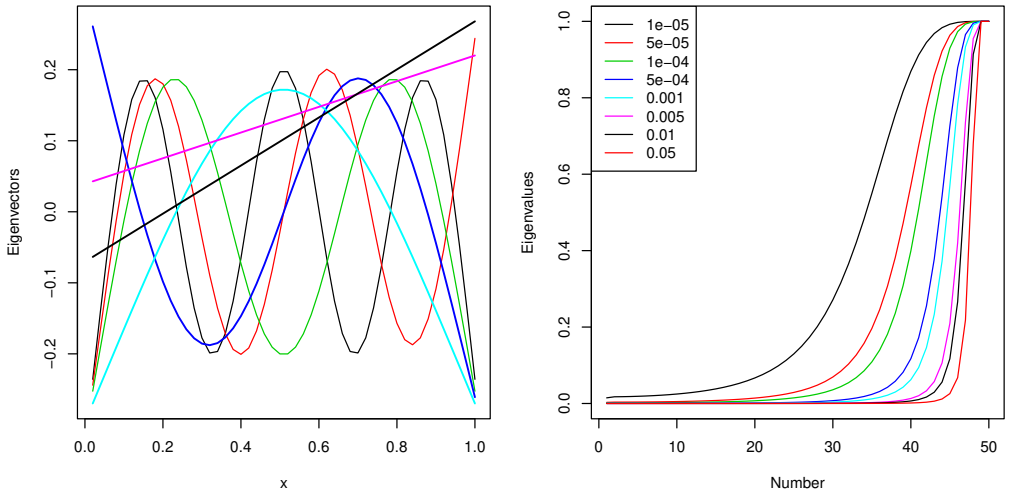


Figure 7: *Eigenvectors and eigenvalues for the Reinsch form of the cubic smoothing spline operator, defined over $n = 50$ evenly spaced inputs on $[0, 1]$. The left plot shows the bottom 7 eigenvectors of the Reinsch matrix Q. We can see that the smaller the eigenvalue, the "smoother" the eigenvector. The right plot shows the weights $w_j = 1/(1+\lambda d_j)$, $j = 1, \ldots, n$ 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 $\widehat{\mu} = Sy$, i.e.,

$$\widehat{\mu} = \sum_{j=1}^{n} \frac{u_j^T y}{1 + \lambda d_j} u_j. \tag{32}$$

  Interpretation: smoothing splines perform a regression on the orthonormal basis $u_1, \ldots, u_n \in \mathbb{R}^n$, 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 $u_1, \ldots, u_n$? 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

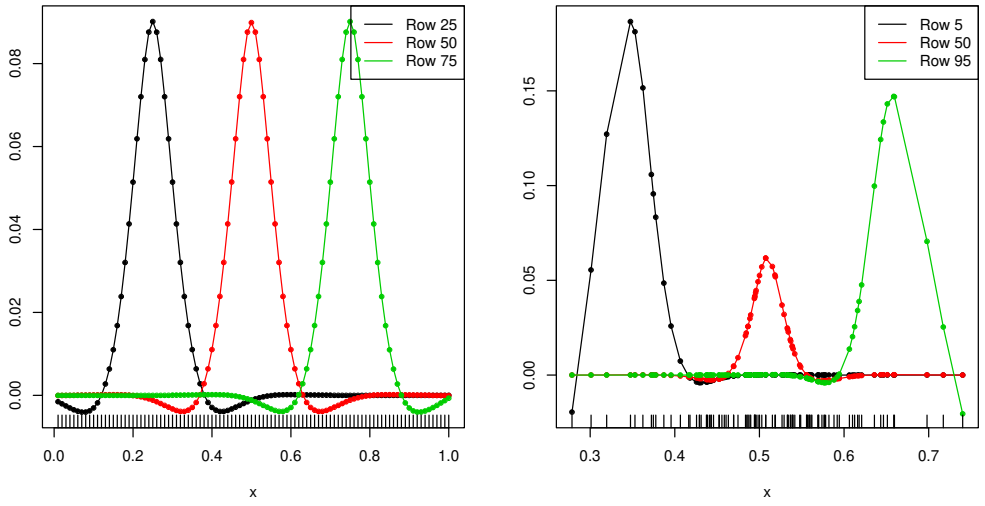


Figure 8: *Rows of the cubic smoothing spline operator S defined over $n = 100$ evenly spaced input points on $[0, 1]$. The left plot shows 3 rows of S (in particular, rows 25, 50, and 75) for $\lambda = 0.0002$. These look precisely like translations of a kernel. The right plot considers a setup where 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":

$$K(x) = \frac{1}{2} \exp(-|x|/\sqrt{2}) \sin(|x|/\sqrt{2} + \pi/4), \tag{33}$$

  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

$$\int K(x)\,dx = 1, \quad \int x^j K(x)\,dx = 0, \quad j = 1, \ldots, 3, \quad \text{but} \quad \int x^4 K(x)\,dx = -24.$$

  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., **??**.

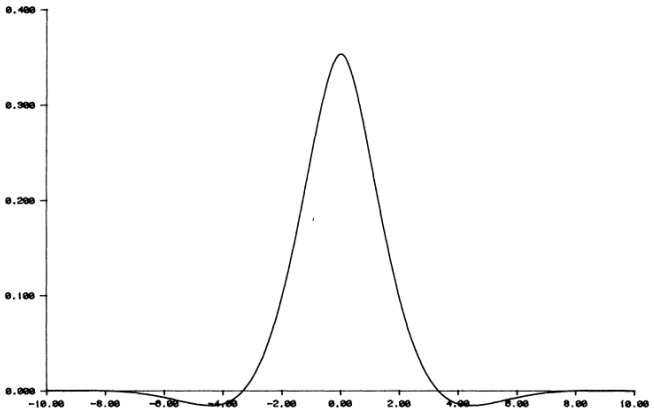


Figure 9: *The Silverman kernel in (33), which is the (asymptotically) equivalent implicit kernel used by smoothing splines. Note that it can be negative. From* **?**

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 $f : \mathbb{R} \to \mathbb{R}$ such that

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

  (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 $\widehat{f}$ in (38) of polynomial order $k = 2m - 1$ with tuning parameter $\lambda \asymp n^{1/(2m+1)} \asymp n^{1/(k+2)}$ satisfies

$$\|\widehat{f} - f_0\|_n^2 \lesssim n^{-2m/(2m+1)} \quad \text{in probability.}$$

  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 $W_1(m, C)$ (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

$$\sum_{\alpha_1+\ldots+\alpha_d=m} \int \left| \frac{\partial^m f(x)}{\partial x_1^{\alpha_1} x_2^{\alpha_2} \ldots \partial x_d^{\alpha_d}} \right|^2 dx,$$

  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 (argurably) 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 nonparametric 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 apadtive 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*, Cambdrige 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

$$\widehat{m} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \operatorname{TV}(f^{(k)}). \tag{34}$$

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 kth 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 kth-order spline are generally not known, i.e., they need not coincide with the inputs $x_1, \ldots, x_n$. (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

$$\widehat{m} = \underset{f \in \mathcal{G}_k}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \operatorname{TV}(f^{(k)}), \tag{35}$$

i.e., we restrict the domain of minimization to be $\mathcal{G}_k$, the space of kth-order spline functions with knots in T_k, where T_k is a subset of $\{x_1, \ldots, x_n\}$ 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 kth-order splines with knots in T_k has dimension $|T_k| + k + 1 = n$. Therefore we can choose a basis $g_1, \ldots, g_n$ for the functions in $\mathcal{G}_k$, and the problem in (35) becomes one of finding the coefficients in this basis expansion,

$$\widehat{\beta} = \underset{f \in \mathcal{G}_k}{\operatorname{argmin}} \sum_{i=1}^{n} \Big(Y_i - \sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^2 + \lambda \operatorname{TV}\Big\{\Big(\sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^{(k)}\Big\}, \tag{36}$$

and then we have $\widehat{m}(x) = \sum_{j=1}^{n} \widehat{\beta}_j g_j(x)$

Now define the basis matrix $G \in \mathbb{R}^{n \times n}$ by

$$G_{ij} = g_j(X_i), \quad i = 1, \ldots, n.$$

Suppose we choose $g_1, \ldots, g_n$ to be the truncated power basis. Denoting $T_k = \{t_1, \ldots, t_{n-k-1}\}$, we compute

$$\Big(\sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^{(k)} = k. + k. \sum_{j=k+2}^{n} \beta_j 1\{x \geq t_{j-k-1}\},$$

and so

$$\operatorname{TV}\Big\{\Big(\sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^{(k)}\Big\} = k. \sum_{j=k+2}^{n} |\beta_j|.$$

Hence the locally adaptive regression spline problem (36) can be expressed as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - G\beta\|_2^2 + \lambda k. \sum_{i=k+2}^{n} |\beta_i|. \tag{37}$$

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

$$\widehat{m} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \int_0^1 \big(f^{(m)}(x)\big)^2 \, dx, \quad \text{where } m = (k+1)/2. \tag{38}$$

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 $x_1, \ldots, x_n$), 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 $x_1, \ldots, x_n$. 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 $\widehat{m}^{\text{lrs}}$, with $\lambda \asymp n^{1/(2k+3)}$, satisfies

$$\|\widehat{m}^{\text{lrs}} - m_0\|_n^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{in probability},$$

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 $\mathcal{G}_k = \text{span}\{g_1, \ldots, g_n\}$, and $g_1, \ldots, g_n$ are the kth-order truncated power basis

$$\begin{gathered} g_1(x) = 1, \ g_2(x) = x, \ \ldots \ g_{k+1}(x) = x^k, \\ g_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \ldots, p. \end{gathered} \tag{39}$$

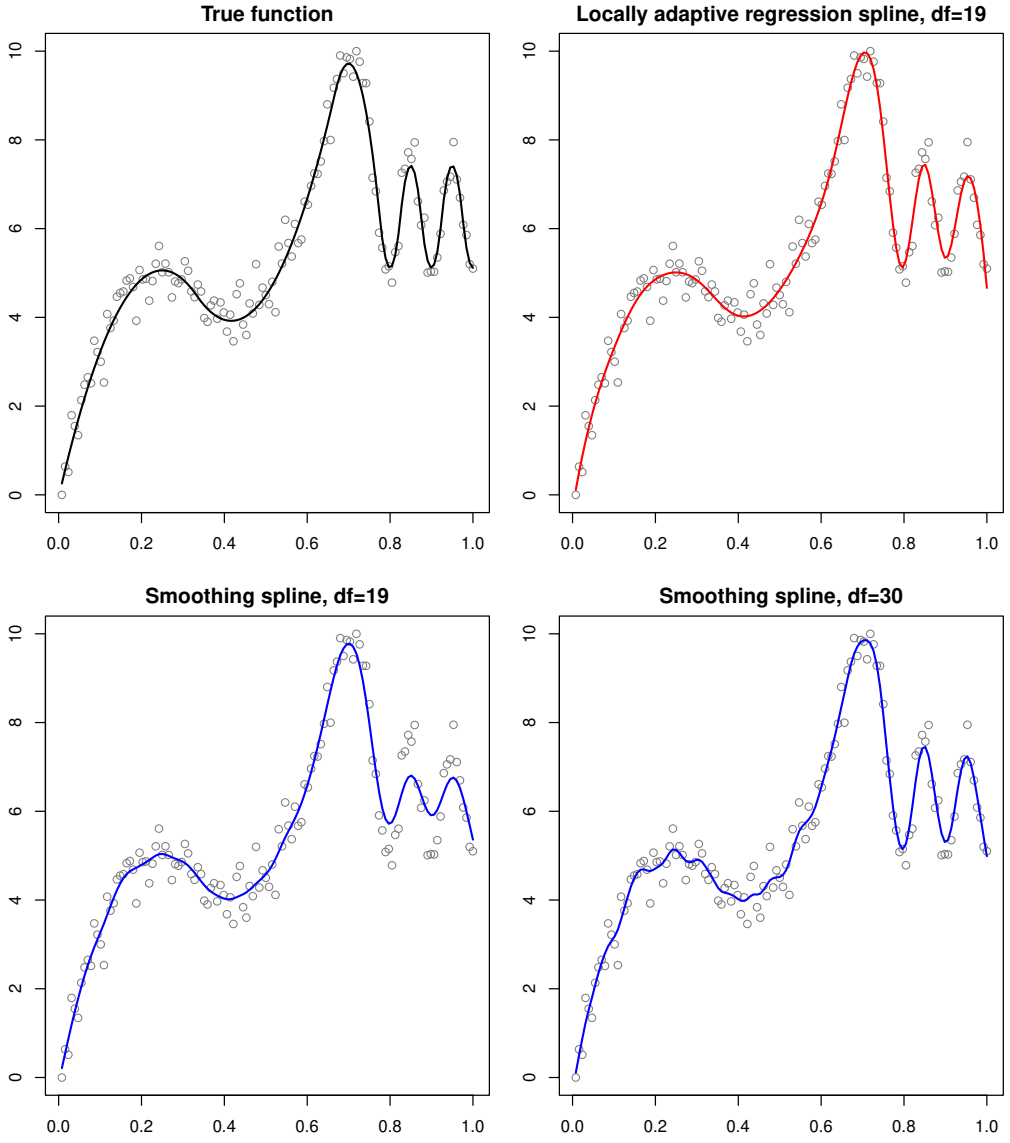


Figure 10: *The top left plot shows a simulated true regression function, which has inhomogeneous smoothness: smoother towards the left part of the domain, wigglier towards the right. The top right plot shows the locally adaptive regression spline estimate with 19 degrees of freedom; notice that it picks up the right level of smoothness throughout. The bottom left plot 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 $T_k \subseteq \{X_1, \ldots X_n\}$ with size $|T_k| = n - k - 1$. The *trend filtering* problem is given by replacing $\mathcal{G}_k$ with a different function space,

$$\widehat{m} = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \operatorname{TV}(f^{(k)}), \tag{40}$$

where the new domain is $\mathcal{H}_k = \operatorname{span}\{h_1, \ldots, h_n\}$. Assuming that the input points are ordered, $x_1 < \ldots < x_n$, the functions $h_1, \ldots, h_n$ are defined by

$$\begin{aligned} h_j(x) &= \prod_{\ell=1}^{j-1} (x - x_\ell), \quad j = 1, \ldots, k+1, \\ h_{k+1+j}(x) &= \prod_{\ell=1}^{k} (x - x_{j+\ell}) \cdot 1\{x \geq x_{j+k}\}, \quad j = 1, \ldots, n-k-1. \end{aligned} \tag{41}$$

(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 rth-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 $(x - x_{j+k})(x - x_{j+k-1}) \cdot \ldots, (x - x_{j+1})$

Defining the falling factorial basis matrix

$$H_{ij} = h_j(X_i), \quad i, j = 1, \ldots, n,$$

it is now straightforward to check that the proposed problem of study, trend filtering in (40), is equivalent to

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - H\beta\|_2^2 + \lambda k. \sum_{i=k+2}^{n} |\beta_i|. \tag{42}$$

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 $h_1, \ldots, h_n$ 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

$$\widehat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - \theta\|_2^2 + \lambda \|D\theta\|_1, \tag{43}$$

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 $X_i = i$, $i = 1, \ldots, n$. Then, e.g., the first 3 orders of difference operators are:

$$D = \begin{bmatrix} -1 & 1 & 0 & \ldots \\ 0 & -1 & 1 & \ldots \\ \vdots & & & \end{bmatrix}, \quad D = \begin{bmatrix} 1 & -2 & 1 & 0 & \ldots \\ 0 & 1 & -2 & 1 & \ldots \\ 0 & 0 & 1 & -2 & \ldots \\ \vdots & & & & \end{bmatrix}, \quad D = \begin{bmatrix} -1 & 3 & -3 & 1 & \ldots \\ 0 & -1 & 3 & -3 & \ldots \\ 0 & 0 & -1 & 3 & \ldots \\ \vdots & & & & \end{bmatrix}$$

when $k = 0$ when $k = 1$ when $k = 2$.

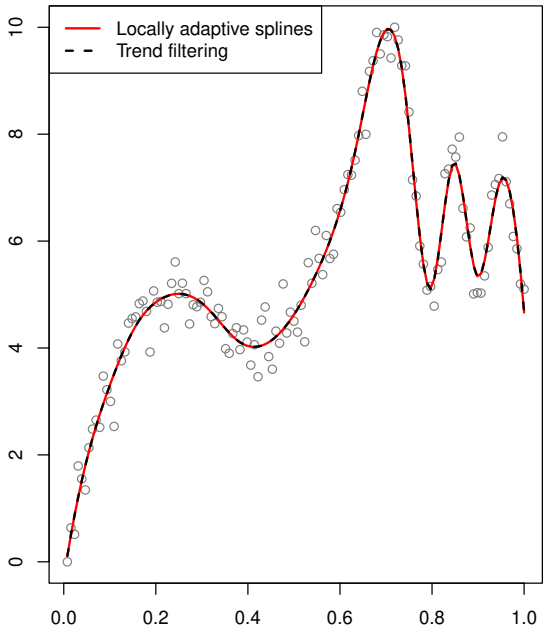


Figure 11: *Trend filtering and locally adaptive regression spline estimates, fit on the same data set as in Figure 10. The two are tuned at the same level, and the estimates are visually indistinguishable*

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 $\widehat{m}^{\mathrm{tf}}$ and $\widehat{m}^{\mathrm{lrs}}$, respectively, when the tuning parameter λ for each scales as $n^{1/(2k+3)}$, satisfy

$$\|\widehat{m}^{\mathrm{tv}} - \widehat{m}^{\mathrm{lrs}}\|_n^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{in probability.}$$

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,

$$\mathbb{E}\left((\widehat{m}_h(x) - m_h(x))^2 \,\middle|\, X_1, \ldots, X_n \right) = \frac{\sum_{i=1}^n \mathrm{Var}(Y_i|X_i) I(\|X_i - x\| \leq h)}{n^2 P_n^2(B(x,h))} \leq \frac{\sigma^2}{nP_n(B(x,h))}.$$

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,

$$\begin{aligned}
\mathbb{E}\int (\widehat{m}_h(x) - m(x))^2 dP(x) &= \mathbb{E}\int (\widehat{m}_h(x) - m_h(x))^2 dP(x) + \mathbb{E}\int (m_h(x) - m(x))^2 dP(x) \\
&\leq \mathbb{E}\int \frac{\sigma^2}{nP_n(B(x,h))} I_{A_n(x)} dP(x) + L^2 h^2 + \int m^2(x) \mathbb{E}(I_{A_n(x)^c}) dP(x). \qquad (44)
\end{aligned}$$

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 $z_1, \ldots, z_M$ 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

$$\begin{aligned}
\int m^2(x)\mathbb{E}(I_{A_n(x)^c})dP(x) &\leq \sup_x m^2(x)\int (1-P(B(x,h)))^n dP(x) \\
&\leq \sup_x m^2(x)\int e^{-nP(B(x,h))}dP(x) \\
&= \sup_x m^2(x)\int e^{-nP(B(x,h))}\frac{nP(B(x,h))}{nP(B(x,h))}dP(x) \\
&\leq \sup_x m^2(x)\sup_u(ue^{-u})\int \frac{1}{nP(B(x,h))}dP(x) \\
&\leq \sup_x m^2(x)\sup_u(ue^{-u})\frac{c_1}{nh^d} = \frac{c_2}{nh^d}.
\end{aligned}$$

10.13 Proof of the Spline Lemma

The key result can be stated as follows: if $\widetilde{f}$ is any twice differentiable function on $[0,1]$, and $x_1,\ldots,x_n \in [0,1]$, then there exists a natural cubic spline f with knots at $x_1,\ldots,x_n$ such that $m(X_i) = \widetilde{f}(X_i)$, $i = 1,\ldots,n$ and

$$\int_0^1 f''(x)^2\,dx \leq \int_0^1 \widetilde{f}''(x)^2\,dx.$$

Note that this would in fact prove that we can restrict our attention in (25) to natural splines with knots at $x_1,\ldots,x_n$.

The natural spline basis with knots at $x_1,\ldots,x_n$ is n-dimensional, so given any n points $z_i = \widetilde{f}(X_i)$, $i = 1,\ldots,n$, we can always find a natural spline f with knots at $x_1,\ldots,x_n$ that satisfies $m(X_i) = z_i$, $i = 1,\ldots,n$. Now define

$$h(x) = \widetilde{f}(x) - m(x).$$

Consider

$$\begin{aligned}
\int_0^1 f''(x)h''(x)\,dx &= f''(x)h'(x)\Big|_0^1 - \int_0^1 f'''(x)h'(x)\;dx \\
&= -\int_{x_1}^{x_n} f'''(x)h'(x)\;dx \\
&= -\sum_{j=1}^{n-1} f'''(x)h(x)\Big|_{x_j}^{x_{j+1}} + \int_{x_1}^{x_n} f^{(4)}(x)h'(x)\;dx \\
&= -\sum_{j=1}^{n-1} f'''(x_j^+)\big(h(x_{j+1}) - h(x_j)\big),
\end{aligned}$$

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}), $j = 1,\ldots,n-1$, 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 $j = 1, \ldots, n$, we have

$$\int_0^1 f''(x) h''(x)\, dx = 0.$$

From this, it follows that

$$\begin{aligned}
\int_0^1 \widetilde{f}''(x)^2\, dx &= \int_0^1 \big(f''(x) + h''(x)\big)^2\, dx \\
&= \int_0^1 f''(x)^2\, dx + \int_0^1 h''(x)^2\, dx + 2\int_0^1 f''(x) h''(x)\, dx \\
&= \int_0^1 f''(x)^2\, dx + \int_0^1 h''(x)^2\, dx,
\end{aligned}$$

and therefore

$$\int_0^1 f''(x)^2\, dx \leq \int_0^1 \widetilde{f}''(x)^2\, dx, \tag{45}$$

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 $j = 1, \ldots, n$, this is equivalent to $h = 0$. In other words, the inequality (45) holds strictly except when $\widetilde{f} = f$, so the solution in (25) is uniquely a natural spline with knots at the inputs.

Linear Regression

We observe $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X_i = (X_i(1), \ldots, X_i(d)) \in \mathbb{R}^d$ 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

$$R(\widehat{m}) = \mathbb{E}[(Y - \widehat{m}(X))^2 | \mathcal{D}] = \int (y - \widehat{m}(x))^2 dP(x, y)$$

and the *prediction risk* of $\widehat{m}$ is

$$r(\widehat{m}) = \mathbb{E}(Y - \widehat{m}(X))^2 = \mathbb{E}[r(\widehat{m})]$$

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:

$$r(\widehat{m}) = \sigma^2 + \int b_n^2(x) dP(x) + \int v_n(x) dP(x)$$

where

$$\sigma^2 = \mathbb{E}[Y - m(X)]^2, \quad b_n(x) = \mathbb{E}[\widehat{m}(x)] - m(x), \quad v_n(x) = \mathrm{Var}(\widehat{m}(x)).$$

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

$$\widehat{r}_n(\beta) = \frac{1}{n} \sum_i (Y_i - X_i^T \beta)^2$$

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 $\epsilon > 0$,*

$$\sup_{P\in\mathcal{P}} P^n\Big(r(\widehat{\beta}_n) > r(\beta_*(P)) + 2\epsilon\Big) \leq c_1 e^{-nc_2\epsilon^2}. \tag{2}$$

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,

$$P\Big(\sup_{\beta\in B} |\widehat{r}_n(\beta) - r(\beta)| > \epsilon\Big) \leq c_1 e^{-nc_2\epsilon^2}.$$

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* $\sigma^2=\sup_x \mathrm{Var}(Y|X=x)<\infty$. *Assume that all the random variables are bounded by* $L<\infty$. *Then*

$$\mathbb{E}\int|\widehat{\beta}^Tx-m(x)|^2dP(x)\leq 8\inf_\beta\int|\beta^Tx-m(x)|^2dP(x)+\frac{Cd(\log(n)+1)}{n}.$$

The proof is straightforward but is very long. The strategy is to first bound $n^{-1}\sum_i(\widehat{\beta}^TX_i-m(X_i))^2$ 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

$$\Gamma=\Sigma^{-1}\mathbb{E}\Big[(Y-X^T\beta)^2XX^T\Big]\Sigma^{-1}$$

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 $\widehat{\epsilon}_i=Y_i-\widehat{\beta}^TX_i$.

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 the 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 the 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 $\{1, \ldots, d\}$ 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 approxmation 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 corrleation between Y and the j^{th} feature. Let $J = \text{argmax}_j |r_j|$. Let $S = S \cup \{J\}$.
3. Compute the regression of Y on $X_S = (X(j) : \ j \in S)$. Compute the residuals $e = (e_1, \ldots, e_n)$ where $e_i = Y_i - \widehat{\beta}_S^T X_i$.
4. Compute the correlations r_j between the residuals e and the remaining features.
5. Let $J = \text{argmax}_j |r_j|$. 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

$$\widehat{\beta} = (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\alpha}.$$

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 apprximation to $m(x)$. Then, with probability at least $1 - 4e^{-t}$,*

$$r(\widehat{\beta}) - r(\beta) \leq \left(1 + O\left(\frac{1 + \frac{r^2}{\lambda}}{n}\right)\right) \frac{\lambda ||\beta||^2}{2} + \frac{\sigma^2}{n} \frac{\text{tr}(\Sigma)}{2\lambda}.$$

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 Fugure 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 $\mathcal{D} = \{\psi_1, \psi_2, \ldots, \}$. 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

$$g_N = \text{argmax}_{\psi \in \mathcal{D}} |\langle r_{N-1}, \psi \rangle|$$

and set $V_N = V_{N-1} \cup \{g_N\}$. Let f_N be the projection of r_{N-1} onto $\text{Span}(V_N)$. 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

$$\sigma_N(f) = \inf_{|\Lambda| \leq N} \inf_{g \in \text{Span}(\Lambda)} \|f - g\| \tag{3}$$

where Λ denotes a subset of $\mathcal{D}$ and $\text{Span}(\Lambda)$ 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 satsifies*

$$\|r_N\| \leq \frac{\|f\|_{\mathcal{L}_1}}{\sqrt{N+1}} \tag{4}$$

for all $N \geq 1$.

Proof. Note that f_N is the best approximation to f from $\text{Span}(V_N)$. 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 $a_0 \geq a_1 \geq a_2 \geq \cdots$ 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 $M = \|f\|_{\mathcal{L}_1}^2$. □

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,

$$|\langle r_{N-1}, g_k\rangle|^2 \geq \frac{(\|r_{N-1}\|^2 - \|f - h\|^2)^2}{4\|h\|^2_{\mathcal{L}_1}}.$$

Thus,

$$a_N \leq a_{N-1}\left(1 - \frac{a_{N-1}}{4\|h\|^2_{\mathcal{L}_1}}\right)$$

where $a_N = \|r_N\|^2 - \|f - h\|^2$. By induction, the last displayed inequality implies that $a_N \leq 4\|h\|^2_{\mathcal{L}_1}/k$ and the result follows. □

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 $Y = (Y_1, \ldots, Y_n)^T$ 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: $r_0 = Y$, $\widehat{f}_0 = 0$, $V = \emptyset$.

3. Repeat: At step N define

$$g_N = \text{argmax}_{\psi \in \mathcal{D}} |\langle r_{N-1}, \psi \rangle_n|$$

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 Span(V_N). Let $r_N = Y - f_N$.

Figure 3: The Greedy (Forward Stepwise) Regression Algorithm: Dictionary Version

Theorem 9 *Let* $h_n = \text{argmin}_{h \in \mathcal{F}_N} \|f_0 - h\|^2$. *Suppose that* $\limsup_{n \to \infty} \|h_n\|_{\mathcal{L}_{1,n}} < \infty$. *Let* $N \sim \sqrt{n}$. *Then, for every* $\gamma > 0$, *there exist* $C > 0$ *such that*

$$\|f - \widehat{f}_N\|^2 \leq 4\sigma_N^2 + \frac{C \log n}{n^{1/2}}$$

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

$$\|f - \widehat{f}_L\|^2 \leq \|f - f_L\|^2 + O_P\left(\frac{\log n}{n}\right)^{1/2}$$

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 $\widehat{\beta}$ 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 $\widehat{\beta}$ 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 $\beta = (\beta_1, \ldots, \beta_s, 0, \ldots, 0)$ and decompose the design matrix as $\mathbb{X} = (\mathbb{X}_S \ \mathbb{X}_{S^c})$ where $S = \{1, \ldots, s\}$. Let $\beta_S = (\beta_1, \ldots, \beta_s)$.

Theorem 10 (Sparsistency) *Suppose that:*

1. *The true model is linear.*
2. *The design matrix satisfies*

$$\|\mathbb{X}_{S^c}\mathbb{X}_S(\mathbb{X}_S^T\mathbb{X}_S)^{-1}\|_\infty \leq 1 - \epsilon \quad \text{for some } 0 < \epsilon \leq 1. \tag{7}$$

3. $\phi_n(d_n) > 0$.
4. *The ϵ_i are Normal.*
5. λ_n *satisfies*

$$\frac{n\lambda_n^2}{\log(d_n - s_n)} \to \infty$$

and

$$\frac{1}{\min_{1\leq j\leq s_n}|\beta_j|}\left(\sqrt{\frac{\log s_n}{n}} + \lambda_n \left\|\left(\frac{1}{n}\mathbb{X}^T\mathbb{X}\right)^{-1}\right\|_\infty\right) \to 0. \tag{8}$$

Then the lasso is sparsistent, meaning that $P(\text{support}(\widehat{\beta}) = \text{support}(\beta)) \to 1$ *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.*

[1]The norm $||\beta||_1$ can be thought of as a measure of sparsity. For example, the vectors $x = (1/\sqrt{d}, \ldots, 1/\sqrt{d})$ and $y = (1, 0, \ldots, 1)$ 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. $\liminf_{n\to\infty} \phi_n(s_n \log n) > 0$ *where* $s_n = \|\beta_n\|_0$.

Then

$$\|\beta_n - \widehat{\beta}_n\|^2 = O_P\left(\frac{\log n}{n}\frac{s_n \log n}{\phi_n^2(s_n \log n)}\right) + O\left(\frac{1}{\log n}\right) \tag{9}$$

If

$$s_n \log d_n \left(\frac{\log n}{n}\right) \to 0 \tag{10}$$

and

$$\lambda_n = \sqrt{\frac{\sigma_y^2 \Phi_n(\min n, d_n) n^2}{s_n \log n}} \tag{11}$$

then $\|\widehat{\beta}_n - \beta_n\|^2 \xrightarrow{P} 0$.

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* $\widehat{\beta}$ *be the lasso estimator:*

$$\widehat{\beta} = \underset{||\beta||_1 \leq L}{\operatorname{argmin}}\, \widehat{r}(\beta)$$

where $\widehat{r}(\beta) = n^{-1}\sum_{i=1}^n (Y_i - X_i^T\beta)^2$. *With probabilty at least* $1 - \delta$,

$$r(\widehat{\beta}) \leq r(\beta_*) + \sqrt{\frac{16(L+1)^4 B^2}{n} \log\left(\frac{\sqrt{2}\, d}{\sqrt{\delta}}\right)}.$$

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

$$\widehat{r}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i^T\beta)^2 = \gamma^T \widehat{\Lambda}\gamma$$

where $\widehat{\Lambda} = \frac{1}{n}\sum_{i=1}^n Z_i Z_i^T$. For any $\beta \in \mathcal{B}$,

$$\begin{aligned} |\widehat{r}(\beta) - r(\beta)| &= |\gamma^T(\widehat{\Lambda} - \Lambda)\gamma| \\ &\leq \sum_{j,k} |\gamma(j)|\,|\gamma(k)|\ |\widehat{\Lambda}(j,k) - \Lambda(j,k)| \leq ||\gamma||_1^2 \delta_n \\ &\leq (L+1)^2 \Delta_n \end{aligned}$$

where

$$\Delta_n = \max_{j,k} |\widehat{\Lambda}(j,k) - \Lambda(j,k)|.$$

So,

$$r(\widehat{\beta}) \leq \widehat{r}(\widehat{\beta}) + (L+1)^2\Delta_n \leq \widehat{r}(\beta_*) + (L+1)^2\Delta_n \leq r(\beta_*) + 2(L+1)^2\Delta_n.$$

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,

$$r(\widehat{\beta}) \leq r(\beta_*) + \sqrt{\frac{16(L+1)^4B^2}{n}\log\left(\frac{\sqrt{2}\,d}{\sqrt{\delta}}\right)}.$$

with probability at least $1 - \delta$. □

Problems With Sparsity. Sparse estimators are convenient and popular but they can some problems. Say that $\widehat{\beta}$ is **weakly sparsistent** if, for every β,

$$P_\beta\big(I(\widehat{\beta}_j = 1) \leq I(\beta_j = 1) \ \text{ for all } j\big) \to 1 \tag{12}$$

as $n \to \infty$. In particular, if $\widehat{\beta}_n$ is sparsistent, then it is weakly sparsistent. Suppose that d is fixed. Then the least squares estimator $\widehat{\beta}_n$ is minimax and satisfies

$$\sup_\beta E_\beta(n\|\widehat{\beta}_n - \beta\|^2) = O(1). \tag{13}$$

But sparsistent estimators have much larger risk:

Theorem 13 (Leeb and Pötscher (2007)) *Suppose that the following condiitons hold:*

1. *d is fixed.*
2. *The covariariates are nonstochastic and $n^{-1}\mathbb{X}^T\mathbb{X} \to Q$ 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 $\widehat{\beta}$ is weakly sparsistent then

$$\sup_\beta E_\beta(n\|\widehat{\beta}_n - \beta\|^2) \to \infty. \tag{14}$$

More generally, if ℓ is any nonnegative loss function then

$$\sup_\beta E_\beta(\ell(n^{1/2}(\widehat{\beta}_n - \beta))) \to \sup_s \ell(s). \tag{15}$$

Proof. Choose any $s \in \mathbb{R}^d$ and let $\beta_n = -s/\sqrt{n}$. Then,

$$\begin{aligned}\sup_\beta E_\beta(\ell(n^{1/2}(\widehat{\beta} - \beta)) &\geq E_{\beta_n}(\ell(n^{1/2}(\widehat{\beta} - \beta)) \geq E_{\beta_n}(\ell(n^{1/2}(\widehat{\beta} - \beta))I(\widehat{\beta} = 0)) \\ &= \ell(-\sqrt{n}\beta_n)P_{\beta_n}(\widehat{\beta} = 0) = \ell(s)P_{\beta_n}(\widehat{\beta} = 0).\end{aligned}$$

Now, $P_0(\widehat{\beta} = 0) \to 1$ by assumption. It can be shown that we also have $P_{\beta_n}(\widehat{\beta} = 0) \to 1$.[2] Hence, with probability tending to 1,

$$\sup_\beta E_\beta(\ell(n^{1/2}(\widehat{\beta} - \beta)) \geq \ell(s).$$

Since s was arbitrary the result follows. □

It follows that, if R_n denotes the minimax risk then

$$\sup_\beta \frac{R(\widehat{\beta}_n)}{R_n} \to \infty.$$

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.

[2]This 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 $m_{\widehat{h}}$ 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 $m_{\widehat{h}}$.

Split the data into training $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ and test $\mathcal{D}' = \{(X_1', Y_1'), \ldots, (X_n', Y_n')\}$. 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 $\delta > 0$,*

$$\mathbb{E} \int |m_H(x) - m(x)|^2 dP(x) \leq (1+\delta)\mathbb{E} \int |m_{\widehat{h}}(x) - m(x)|^2 dP(x) + \frac{C(1 + \log|M|)}{n}$$

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

$$T_2 = (1+\delta)\frac{1}{n} \sum_{\mathcal{D}'} \left(|Y_i - m_H(X_i)|^2 - |Y_i - m(X_i)|^2 \right) \leq (1+\delta)\frac{1}{n} \sum_{\mathcal{D}'} \left(|Y_i - m_{\widehat{h}}(X_i)|^2 - |Y_i - m(X_i)|^2 \right)$$

and so

$$\begin{aligned} \mathbb{E}[T_2 | \mathcal{D}] &\leq (1+\delta) \left(\mathbb{E}(|Y - m_{\widehat{h}}(X)|^2 | \mathcal{D}) - \mathbb{E}|Y - m(X)|^2 \right) \\ &= (1+\delta) \int |m_{\widehat{h}}(x) - m(x)|^2 dP(x). \end{aligned}$$

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

$$(1+\delta) \left(\mathbb{E}(|m_H(X) - Y|^2 | \mathcal{D}) - \mathbb{E}|m(X) - Y|^2 - \frac{1}{n} \sum_{\mathcal{D}'} \left(|Y_i - m_H(X_i)|^2 - |Y_i - m(X_i)|^2 \right) \right) \geq$$
$$s + \delta \left(\mathbb{E}|m_H(X) - Y|^2 - \mathbb{E}|m(X) - Y|^2 \right).$$

This has probability at most $|\mathcal{M}|$ times the probabilty that

$$(1+\delta)\left(\mathbb{E}(|m_h(X)-Y|^2|\mathcal{D})-\mathbb{E}|m(X)-Y|^2-\frac{1}{n}\sum_{\mathcal{D}'}\left(|Y_i-m_H(X_i)|^2-|Y_i-m(X_i)|^2\right)\right)\geq$$
$$s+\delta\left(\mathbb{E}|m_h(X)-Y|^2-\mathbb{E}|m(X)-Y|^2\right)$$

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

$$\sigma^2=\mathrm{Var}(Z|\mathcal{D})\leq\mathbb{E}[Z^2|\mathcal{D}]\leq 16L^2\int|m_h(x)-m(x)|^2dP(x)=16L^2\mathbb{E}[Z|\mathcal{D}].$$

Using this, and Bernstein's inequality,

$$\begin{aligned}&P\left(\mathbb{E}[Z|\mathcal{D}]-\overline{Z}\geq\frac{s+\delta\mathbb{E}[Z|\mathcal{D}]}{1+\delta}|\mathcal{D}\right)\\&\leq P\left(\mathbb{E}[Z|\mathcal{D}]-\overline{Z}\geq\frac{s+\delta\sigma^2/(16L^2)}{1+\delta}|\mathcal{D}\right)\\&\leq e^{-nA/B}\end{aligned}$$

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 $u=c\log|\mathcal{M}|/n$. □

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 Apprximations — 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 $\widehat{m}_n^{(0)}, \ldots, \widehat{m}_n^{(k)}, \ldots$, 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: $\widehat{\beta}_j = \sum_i U_i X_{ij} / \sum_i X_{ij}^2$, $j = 1, \ldots, d$.
3. Find $J = \text{argmin}_j RSS_j$ 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 $\widehat{m}^{(k+1)}(x) = \widehat{m}^{(k)}(x) + \nu \text{sign}(\widehat{\beta}_J) x_J$ 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 $\widehat{Y}^{(k)} = (\widehat{m}^{(k)}(X_1), \ldots, \widehat{m}^{(k)}(X_n))^T$,

$$B_k = I - (I - H_k)(I - H_{k-1}) \cdots (I - H_1), \tag{17}$$

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 $k_n \to \infty$ *such that*

$$\mathbb{E}_X |\widehat{m}_n(X) - m_n(x)|^2 \to 0 \tag{19}$$

as $n \to 0$.

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 $\{g_1, \ldots, g_k\}$. 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*

$$\|R_k(f)\| \leq \frac{B}{\left(1 + \sum_{j=1}^k t_j^2\right)^{1/2}} \tag{20}$$

and for the weak greedy algorithm

$$\|R_k(f)\| \leq \frac{B}{\left(1 + \sum_{j=1}^k t_j^2\right)^{t_k/(2(2+t_k))}}. \tag{21}$$

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

$$\mathbb{P}(|\overline{Z} - \mathbb{E}(Z_j)| > \epsilon) \leq 2\exp\left\{-\frac{1}{2}\frac{n\epsilon^2}{\sigma^2 + M\epsilon/3}\right\}. \tag{22}$$

Hence, the probability that any empirical inner products differ from their functional counterparts is no more than

$$d_n^2 \exp\left\{-\frac{1}{2}\frac{n\epsilon^2}{\sigma^2 + M\epsilon/3}\right\} \to 0 \tag{23}$$

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 $\widehat{f}_n$ 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 $\widehat{f}_N$ 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 $h \in \mathcal{D}_n$,*

$$\mathbb{P}\left(\|f - \widehat{f}_N\|^2 > 4\|f - h\|^2 + \frac{8\|h\|^2_{\mathcal{L}_{1,n}}}{N} + \frac{CN\log n}{n}\right) < \frac{1}{n^\gamma}$$

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 $\Lambda \subset \mathcal{D}_n$,*

$$N_1(t, S_\Lambda) \leq 3\left(\frac{2eB}{t}\log\left(\frac{3eB}{t}\right)\right)^{|\Lambda|+1}, \qquad N_2(t, S_\Lambda) \leq 3\left(\frac{2eB^2}{t^2}\log\left(\frac{3eB^2}{t^2}\right)\right)^{|\Lambda|+1}.$$

Also,

$$N_1(t,\mathcal{F}_N) \leq 12p^N\left(\frac{2eB}{t}\log\left(\frac{3eB}{t}\right)\right)^{N+1}, \qquad N_2(t,\mathcal{F}_N) \leq 12p^N\left(\frac{2eB^2}{t^2}\log\left(\frac{3eB^2}{t^2}\right)\right)^{N+1}.$$

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 $\epsilon \in (0, 1/2]$,*

$$\mathbb{P}\left((1-\epsilon)\|f-f_0\|^2 \geq \|Y-f\|_n^2 - \|Y-f_0\|_n^2 + \epsilon(\alpha+\beta) \quad \text{for some } f \in \mathcal{F}\right)$$
$$\leq 14N_1\left(\frac{\beta\epsilon}{20B}, \mathcal{F}\right)\exp\left\{-\frac{\epsilon^2(1-\epsilon)\alpha n}{214(1+\epsilon)B^4}\right\}.$$

Proof of Theorem 18. For any $h \in \mathcal{F}_n$ we have

$$\|\widehat{f}-f_0\|_n^2 = \underbrace{\|\widehat{f}-f_0\|^2 - 2\left(\|Y-\widehat{f}\|_n^2 - \|Y-f_0\|_n^2\right)}_{A_1}$$
$$+\underbrace{2\left(\|Y-\widehat{f}\|_n^2 - \|Y-h\|_n^2\right)}_{A_2} + \underbrace{2\left(\|Y-h\|_n^2 - \|Y-f_0\|_n^2\right)}_{A_3}.$$

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 $\mathrm{Var}(\epsilon_i|X_i) = \sigma^2$. 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:

$$\|\beta\|_0 = \sum_{j=1}^p 1\{\beta_j \neq 0\}, \quad \|\beta\|_1 = \sum_{j=1}^p |\beta_j|, \quad \|\beta\|_2 = \Big(\sum_{j=1}^p \beta_j^2\Big)^{1/2}.$$

(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:

$$\min_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_0 \leq k \qquad \text{(Best subset selection)} \qquad (1)$$

$$\min_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_1 \leq t \qquad \text{(Lasso regression)} \qquad (2)$$

$$\min_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_2^2 \leq t \qquad \text{(Ridge regession)} \qquad (3)$$

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:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_0 \qquad \text{(Best subset selection)} \tag{4}$$

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \qquad \text{(Lasso regression)} \tag{5}$$

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \qquad \text{(Ridge regression)} \tag{6}$$

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 $\widehat{\beta}$ in (2), there is a value of $\lambda \geq 0$ such that $\widehat{\beta}$ 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 $\widehat{\beta}$ in (4), there is a value of $t \geq 0$ such that $\widehat{\beta}$ 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

$$\widehat{\mu} = H(Y; \sqrt{2\lambda}), \quad \widehat{\mu} = S(Y; \lambda), \quad \widehat{\mu} = \frac{Y}{1+2\lambda}$$

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 scontinuous. 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 $\widehat{\beta}$ we will have $\widehat{\beta}_j = 0$ for many components $j \in \{1, \ldots, p\}$. 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

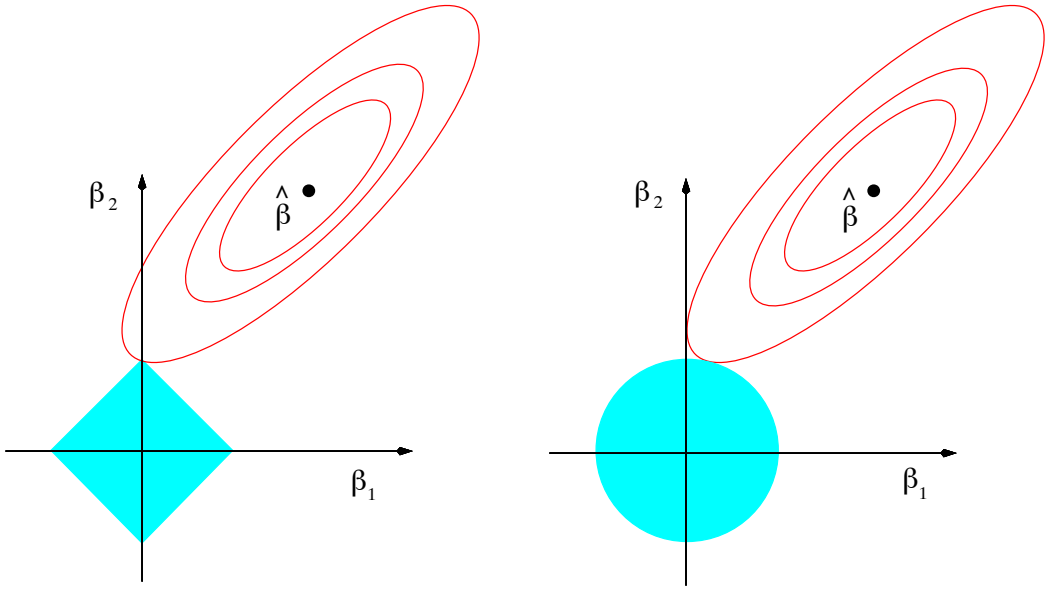


Figure 1: *The "classic" illustration comparing lasso and ridge constraints. From Chapter 3 of Hastie et al. (2009)*

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

$$\widehat{\beta}^{\text{subset}} = H_{\sqrt{2\lambda}}(X^T y), \quad \widehat{\beta}^{\text{lasso}} = S_\lambda(X^T y), \quad \widehat{\beta}^{\text{ridge}} = \frac{X^T y}{1 + 2\lambda}$$

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 exihibits 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 $\widehat{\beta}$ in (4)—meaning that $\widehat{\beta}_j = 0$ for all $j \notin A$, and denoted $A = \text{supp}(\widehat{\beta})$—then we have

$$\begin{aligned} \widehat{\beta}_A &= (X_A^T X_A)^{-1} X_A^T y, \\ \widehat{\beta}_{-A} &= 0. \end{aligned} \tag{7}$$

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.,

$$\begin{aligned} \widehat{\beta}_S^{\text{oracle}} &= (X_S^T X_S)^{-1} X_S^T y, \\ \widehat{\beta}_{-S}^{\text{oracle}} &= 0. \end{aligned}$$

is is called *oracle estimator*, and as we know from our previous calculations, has in-sample risk

$$\frac{1}{n} \|X\widehat{\beta}^{\text{oracle}} - X\beta_0\|_2^2 = \sigma^2 \frac{s_0}{n}.$$

Foster & George (1994) consider this setup, and compare the risk of the best subset selection estimator $\widehat{\beta}$ 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

$$\frac{\mathbb{E}\|X\widehat{\beta} - X\beta_0\|_2^2/n}{\sigma^2 s_0/n} \leq 4 \log p + 2 + o(1), \tag{8}$$

as $n, p \to \infty$. This holds without any conditions on the predictor matrix X. Moreover, they prove the lower bound

$$\inf_{\widehat{\beta}} \sup_{X, \beta_0} \frac{\mathbb{E}\|X\widehat{\beta} - X\beta_0\|_2^2/n}{\sigma^2 s_0/n} \geq 2 \log p - o(\log p),$$

where the infimum is over all estimators $\widehat{\beta}$, 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 presense 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 $\widehat{\beta} = (X^T X + 2\lambda I)^{-1} X^T y$.

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 $\widehat{\beta}$ must satisfy

$$X^T(y - X\widehat{\beta}) = \lambda s, \tag{9}$$

where $s \in \partial\|\widehat{\beta}\|_1$, a subgradient of the ℓ_1 norm evaluated at $\widehat{\beta}$. Precisely, this means that

$$s_j \in \begin{cases} \{+1\} & \widehat{\beta}_j > 0 \\ \{-1\} & \widehat{\beta}_j < 0 \\ [-1, 1] & \widehat{\beta}_j = 0, \end{cases} \quad j = 1, \ldots, p. \tag{10}$$

From (9) we can read off a straightforward but important fact: even though the solution $\widehat{\beta}$ may not be uniquely determined, the optimal subgradient s is a function of the unique fitted value $X\widehat{\beta}$ (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 $\widehat{\beta}$ and $\widetilde{\beta}$ with $\widehat{\beta}_j > 0$ but $\widetilde{\beta}_j < 0$ for some j, and hence we have no problem interpretating 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 $\widehat{\beta}$ is unique. Let $A = \mathrm{supp}(\widehat{\beta})$ be the lasso active set, and let $s_A = \mathrm{sign}(\widehat{\beta}_A)$ be the signs of active coefficients. From the subgradient conditions (9), (10), we know that

$$X_A^T(y - X_A\widehat{\beta}_A) = \lambda s_A,$$

and solving for $\widehat{\beta}_A$ gives

$$\begin{aligned} \widehat{\beta}_A &= (X_A^T X_A)^{-1}(X_A^T y - \lambda s_A), \\ \widehat{\beta}_{-A} &= 0 \end{aligned} \tag{11}$$

(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 $\widehat{\beta}_A$ 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

$$\|\widehat{\beta}\|_1 = s_A^T (X_A^T X_A)^{-1} X_A^T y - \lambda s_A^T (X_A^T X_A)^{-1} s_A < \|(X_A^T X_A)^{-1} X_A^T y\|_1. \tag{12}$$

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 & Buhlmann (2006), Zhao & Yu (2006), Candes & Plan (2009), Wainwright (2009); a helpful text for these kind of results is Buhlmann & 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 $j = 1, \ldots, p$. 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 $\|X^T\epsilon\|_\infty = \max_{j=1,\ldots,p} |X_j^T\epsilon|$ 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 $\delta > 0$,

$$\max_{j=1,\ldots,p} |X_j^T\epsilon| \leq \sigma\sqrt{2n\log(ep/\delta)},$$

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

$$\frac{1}{n}\|X\widehat{\beta} - X\beta_0\|_2^2 \leq 4\sigma\|\beta_0\|_1\sqrt{\frac{2\log(ep/\delta)}{n}}, \tag{13}$$

with probability at least $1-\delta$.

The high-probability result (13) implies an in-sample risk bound of

$$\frac{1}{n}\mathbb{E}\|X\widehat{\beta} - X\beta_0\|_2^2 \lesssim \|\beta_0\|_1\sqrt{\frac{\log p}{n}}.$$

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, $i = 1,\ldots,n$ 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

$$\mathbb{E}(x_0^T\widehat{\beta} - x_0^T\beta)^2 \lesssim \|\beta_0\|_1^2\sqrt{\frac{\log p}{n}}.$$

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 that[1]

$$\langle X^T(y - X\widehat{\beta}), \widetilde{\beta} - \widehat{\beta} \rangle \leq 0. \tag{14}$$

Rearranging gives

$$\langle \mu(X) - X\widehat{\beta}, X\widetilde{\beta} - X\widehat{\beta} \rangle \leq \langle X^T\epsilon, \widehat{\beta} - \widetilde{\beta} \rangle. \tag{15}$$

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 that 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.,

$$\frac{1}{n}\|Xv\|_2^2 \geq \phi_0^2\|v\|_2^2 \quad \text{for all subsets } J \subseteq \{1, \ldots, p\} \text{ such that } |J| = s_0$$
$$\text{and all } v \in \mathbb{R}^p \text{ such that } \|v_{J^c}\|_1 \leq 3\|v_J\|_1 \tag{16}$$

[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, $\langle \nabla f(\widehat{x}), z - \widehat{z} \rangle \geq 0$.

then

$$\|\widehat{\beta} - \beta_0\|_2^2 \lesssim \frac{s_0 \log p}{n\phi_0^2} \tag{17}$$

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 $\|A\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^q |A_{ij}|$ denotes the ℓ_∞ norm of an $m \times q$ matrix A

Under these assumptions, once can show that, if λ is chosen just right, then

$$P(\text{support}(\widehat{\beta}) = \text{support}(\beta)) \to 1. \tag{18}$$

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 presense 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. (2000*a*), '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. (2000*b*), '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. & Buhlmann, 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 $f : \mathbb{R}^n \to \mathbb{R}$ is called *convex* if its domain $\mathrm{dom}(f)$ is convex, and for any $x, y \in \mathrm{dom}(f)$ and $t \in [0, 1]$,

$$f\big(tx + (1 - t)y\big) \leq tf(x) + (1 - t)f(y),$$

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, hyperplans, 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

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \;\; i = 1, \ldots m \\ & \ell_j(x) = 0, \;\; j = 1, \ldots r \end{aligned}$$

Here $D = \mathrm{dom}(f) \cap \bigcap_{i=1}^m \mathrm{dom}(h_i) \cap \bigcap_{j=1}^r \mathrm{dom}(\ell_j)$ is the common domain of all functions. A *convex optimization problem* is an optimization problem in which all functions $f, h_1, \ldots h_m$ are convex, and all functions $\ell_1, \ldots \ell_r$ are affine. (Think: why affine?) Hence, we can express it as

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \;\; i = 1, \ldots m \\ & Ax = b \end{aligned}$$

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,

$$f\big(tz + (1-t)x\big) \leq tf(z) + (1-t)f(x) < f(x)$$

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 it 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 continously drop to zero, and coefficients of variables to enter the active set would continuously move form zero. This would make whole the lasso solution continuous. Fortuitously, this is indeed the case, and the lasso solution $\widehat{\beta}$ 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 $X\widehat{\beta}$ 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

$$X\widehat{\beta} = (I - P_C)(y),$$

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 $A \subseteq \{1, \ldots p\}$ 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 $A \subseteq \{1, \ldots p\}$ 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 that see that as we wiggle y around, it will project to the same face. From the

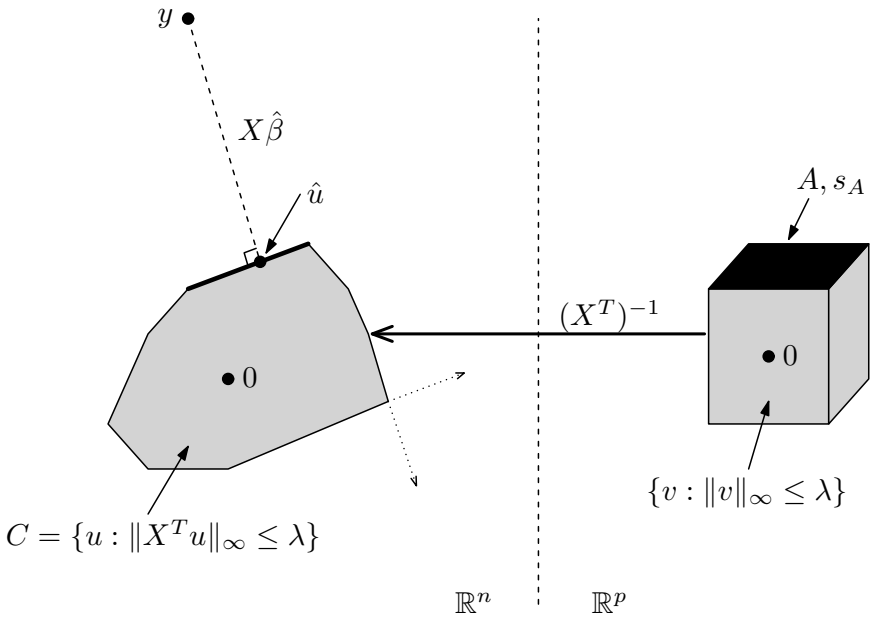


Figure 2: *A geometric picture of the lasso solution. The left panel shows the polyhedron underlying all lasso fits, where each face corresponds to a particular combination of active set A and signs s; the right panel displays the "inverse" polyhedron, where the dual solutions live*

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 $\widehat{\beta}(\lambda)$, 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 $\widehat{\beta}(\lambda)$, $\lambda \in [0, \infty]$ 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 $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r \geq 0$, 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 $\lambda_1 \geq \ldots \geq \lambda_r$ in the solution path correspond to λ values at which the active set $A(\lambda) = \mathrm{supp}(\widehat{\beta}(\lambda))$ 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 variables leaves the active set. See Figure 3.

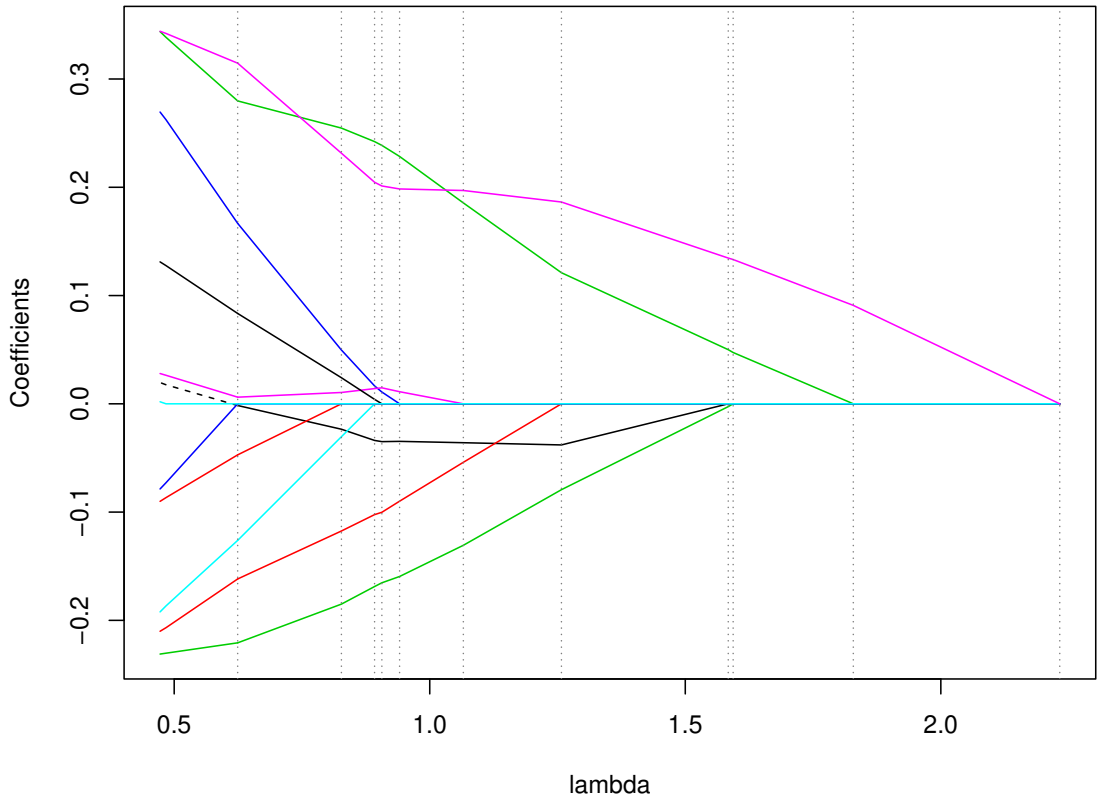


Figure 3: *An example of the lasso path. Each colored line denotes a component of the lasso solution* $\widehat{\beta}_j(\lambda)$, $j = 1, \ldots, p$ *as a function of* λ. *The gray dotted vertical lines mark the knots* $\lambda_1 \geq \lambda_2 \geq \ldots$

The lasso solution path was described by Osborne et al. (2000*a*,*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 $\widehat{\beta}(\infty) = 0$.

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 $\widehat{u}(\lambda) = P_{C_\lambda}(y)$, 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 $X\widehat{\beta}(\lambda) = y - \widehat{u}(\lambda)$ will also behave linearly in λ. Eventually, as we continue to decrease λ, the projected point $\widehat{u}(\lambda)$ 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 $\widehat{\beta}(\infty) = 0$, as indicated above. Consider decreasing λ from ∞, and continuing to set $\widehat{\beta}(\lambda) = 0$ 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 $j = 1, \ldots, p$. 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 $j = 1, \ldots, p$. In short, $\widehat{\beta}(\lambda) = 0$ is the lasso solution for all $\lambda \geq \lambda_1$, where

$$\lambda_1 = \max_{j=1,\ldots,p} |X_j^T y|. \tag{19}$$

What happens next? As we decrease λ from λ_1, we know that we're going to have to change $\widehat{\beta}(\lambda)$ 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 $\widehat{\beta}_{j_1}(\lambda)$ nonzero. Consider setting

$$\begin{aligned} \widehat{\beta}_{j_1}(\lambda) &= (X_{j_1}^T X_{j_1})^{-1}(X_{j_1}^T y - \lambda s_{j_1}) \\ \widehat{\beta}_j(\lambda) &= 0, \quad \text{for all } j \neq j_1, \end{aligned} \tag{20}$$

as λ decreases from λ_1, where $s_{j_1} = \text{sign}(X_{j_1}^T y)$. Note that this makes $\widehat{\beta}(\lambda)$ a piecewise linear and continuous function of λ, so far. The KKT conditions are then

$$X_{j_1}^T \Big(y - X_{j_1}(X_{j_1}^T X_{j_1})^{-1}(X_{j_1}^T y - \lambda s_{j_1}) \Big) = \lambda s_{j_1},$$

which can be checked with simple algebra, and

$$\Big| X_j^T \Big(y - X_{j_1}(X_{j_1}^T X_{j_1})^{-1}(X_{j_1}^T y - \lambda s_{j_1}) \Big) \Big| \leq \lambda,$$

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

$$\lambda_2 = \max_{j \neq j_1,\, s_j \in \{-1,1\}}{}^{+} \frac{X_j^T(I - X_{j_1}(X_{j_1}^T X_{j_1})^{-1}X_{j_1})y}{s_j - X_j^T X_{j_1}(X_{j_1}^T X_{j_1})^{-1}s_{j_1}}, \tag{21}$$

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

$$\begin{aligned} \widehat{\beta}_A(\lambda) &= (X_A^T X_A)^{-1}(X_A^T y - \lambda s_A) \\ \widehat{\beta}_{-A}(\lambda) &= 0, \end{aligned} \tag{22}$$

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

$$\begin{aligned} \widehat{\beta}_A(\lambda) &= \widehat{\beta}_A(\lambda_k) + (\lambda_k - \lambda)(X_A^T X_A)^{-1} s_A, \\ \widehat{\beta}_{-A}(\lambda) &= 0. \end{aligned} \tag{23}$$

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 & Buhlmann (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 $1-\delta$,

$$\|X\widehat{\beta} - X\beta_0\|_2^2 \leq 2\sigma\sqrt{2n\log(ep/\delta)}\|\widehat{\beta} - \beta_0\|_1 + 2\lambda(\|\beta_0\|_1 - \|\widehat{\beta}\|_1).$$

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

$$\begin{aligned}
\|X\widehat{\beta} - X\beta_0\|_2^2 &\leq \lambda\|\widehat{\beta} - \beta_0\|_1 + 2\lambda(\|\beta_0\|_1 - \|\widehat{\beta}\|_1) \\
&\leq \lambda\|\widehat{\beta}_S - \beta_{0,S}\|_1 + \lambda\|\widehat{\beta}_{-S}\|_1 + 2\lambda(\|\beta_0\|_1 - \|\widehat{\beta}\|_1) \\
&\leq \lambda\|\widehat{\beta}_S - \beta_{0,S}\|_1 + \lambda\|\widehat{\beta}_{-S}\|_1 + 2\lambda(\|\beta_{0,S} - \widehat{\beta}_S\|_1 - \|\widehat{\beta}_{-S}\|_1) \\
&= 3\lambda\|\widehat{\beta}_S - \beta_{0,S}\|_1 - \lambda\|\widehat{\beta}_{-S}\|_1,
\end{aligned}$$

where the two inequalities both followed from the triangle inequality, one application for each of the two terms, and we have used that $\widehat{\beta}_{0,-S} = 0$. As $\|X\widehat{\beta} - X\beta_0\|_2^2 \geq 0$, we have shown

$$\|\widehat{\beta}_{-S} - \widehat{\beta}_{0,-S}\|_1 \leq 3\|\widehat{\beta}_S - \beta_{0,S}\|_1,$$

and thus we may apply the compatibility condition (25) to the vector $v = \widehat{\beta} - \beta_0$. 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)

$$\|X\widehat{\beta} - X\beta_0\|_2^2 \leq 3\lambda\|\widehat{\beta}_S - \beta_{0,S}\|_1. \tag{26}$$

For the fitted values, we upper bound the right-hand side of the key inequality (26),

$$\|X\widehat{\beta} - X\beta_0\|_2^2 \leq 3\lambda\sqrt{\frac{s_0}{n\phi_0^2}}\|X\widehat{\beta} - X\beta_0\|_2,$$

or dividing through both sides by $\|X\widehat{\beta} - X\beta_0\|_2$, then squaring both sides, and dividing by n,

$$\frac{1}{n}\|X\widehat{\beta} - X\beta_0\|_2^2 \leq \frac{9s_0\lambda^2}{n^2\phi_0^2}.$$

Plugging in $\lambda = 2\sigma\sqrt{2n\log(ep/\delta)}$, we have shown that

$$\frac{1}{n}\|X\widehat{\beta} - X\beta_0\|_2^2 \leq \frac{72\sigma^2 s_0\log(ep/\delta)}{n\phi_0^2}, \tag{27}$$

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 compability 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 $\|\widehat{\beta} - \beta_0\|_1 \leq 4\|\widehat{\beta}_S - \beta_{0,S}\|_1$,

$$\|\widehat{\beta} - \beta_0\|_1 \leq \frac{12 s_0 \lambda}{n\phi_0^2}.$$

Plugging in $\lambda = 2\sigma\sqrt{2n\log(ep/\delta)}$, we have shown that

$$\|\widehat{\beta} - \beta_0\|_1 \leq \frac{24\sigma s_0}{\phi_0^2}\sqrt{\frac{2\log(ep/\delta)}{n}}, \tag{28}$$

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.,

$$\frac{1}{n}\|Xv\|_2^2 \geq \phi_0^2\|v\|_2^2 \quad \text{for all subsets } J \subseteq \{1,\ldots,p\} \text{ such that } |J| = s_0$$
$$\text{and all } v \in \mathbb{R}^p \text{ such that } \|v_{J^c}\|_1 \leq 3\|v_J\|_1. \tag{29}$$

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 $(1/n)X^TX$; 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 $j = 1,\ldots,p$, 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

$$X_S^T(y - X_S\widehat{\beta}_S) = \lambda s_S \tag{30}$$

$$X_{-S}^T(y - X_S\widehat{\beta}_S) = \lambda s_{-S}. \tag{31}$$

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

$$\widehat{\beta}_S = (X_S^T X_S)^{-1}\big(X_S^T y - \lambda \text{sign}(\beta_{0,S})\big), \tag{32}$$

where we have substituted $s_S = \text{sign}(\beta_{0,S})$. From (31), this implies that s_{-S} must satisfy

$$s_{-S} = \frac{1}{\lambda} X_{-S}^T\big(I - X_S(X_S^T X_S)^{-1}X_S^T\big)y + X_{-S}^T X_S(X_S^T X_S)^{-1}\text{sign}(\beta_{0,S}). \tag{33}$$

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

$$\text{sign}(\beta_{0,j} + \Delta_j) = \text{sign}(\beta_{0,j}), \text{ where}$$
$$\Delta_j = e_j^T (X_S^T X_S)^{-1} \big(X_S^T \epsilon - \lambda \text{sign}(\beta_{0,S}) \big), \text{ for all } j \in S, \quad (34)$$

and

$$\Big\| \frac{1}{\lambda} X_{-S}^T \big(I - X_S (X_S^T X_S)^{-1} X_S^T \big) \epsilon + X_{-S}^T X_S (X_S^T X_S)^{-1} \text{sign}(\beta_{0,S}) \Big\|_\infty < 1. \quad (35)$$

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 simply, 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

$$\sigma^2 \Lambda_{\max} \Big((X_S^T X_S)^{-1} \Big) \leq \frac{\sigma^2 n}{C},$$

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

$$\begin{aligned} \|\Delta_S\|_\infty &\leq \frac{\sigma}{\sqrt{C}} \sqrt{2n \log(es_0/\delta)} + \lambda \|(X_S^T X_S)^{-1}\|_\infty \\ &\leq \beta_{0,\min} + \frac{\gamma}{\sqrt{C}} \underbrace{\left(\frac{\sigma}{\gamma} \sqrt{2n \log(es_0/\delta)} - 4\lambda \right)}_{a}. \end{aligned}$$

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

$$\Big\| \frac{1}{\lambda} X_{-S}^T \big(I - X_S (X_S^T X_S)^{-1} X_S^T \big) \epsilon + X_{-S}^T X_S (X_S^T X_S)^{-1} \text{sign}(\beta_{0,S}) \Big\|_\infty \leq \|z\|_\infty + (1 - \gamma),$$

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

$$\begin{aligned}\Big\|\frac{1}{\lambda}X_{-S}^T\big(I - X_S(X_S^TX_S)^{-1}X_S^T\big)\epsilon + X_{-S}^TX_S(X_S^TX_S)^{-1}\mathrm{sign}(\beta_{0,S})\Big\|_\infty \\ \leq \frac{\sigma}{\lambda}\sqrt{2n\log\left(e(p-s_0)/\delta\right)} + (1-\gamma) \\ = 1 + \underbrace{\left(\frac{\sigma}{\lambda}\sqrt{2n\log\left(e(p-s_0)/\delta\right)} - \gamma\right)}_{b},\end{aligned}$$

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 $(1/n)(X_S^TX_S)^{-1}$ is somewhat like the restricted eigenvalue condition on X. But the mutual incoherence condition is even stronger; it requires the regression coefficients

$$\eta_j(S) = (X_S^TX_S)^{-1}X_S^TX_j,$$

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 $j = 1, \ldots, p$, and $\epsilon \sim N(0, \sigma^2)$, Raskutti et al. (2011) derive upper and lower bounds on the minimax prediction error

$$M(s_0, n, p) = \inf_{\widehat{\beta}} \sup_{\|\beta_0\|_0 \leq s_0} \frac{1}{n} \|X\widehat{\beta} - X\beta_0\|_2^2.$$

(Their analysis is acutally considerably more broad than this and covers the coefficient error $\|\widehat{\beta} - \beta_0\|_2$, 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)

Journal of the Royal Statistical Society
SERIES B Statistical Methodology

J. R. Statist. Soc. B (2009)
71, *Part* 5, *pp.* 1009–1030

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 $Y_i = X_i^{\mathrm{T}}\beta + \varepsilon_i$, for $i = 1, \ldots, n$. 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

$$\sum_i (Y_i - X_i^{\mathrm{T}}\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

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

 1369–7412/09/711009

$$Y_i = \sum_{j=1}^{p} f_j(X_{ij}) + \varepsilon_i. \tag{1}$$

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 garrotte to this setting. As with the parametric non-negative garrotte, 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 $(X_1, Y_1), \ldots, (X_n, Y_n)$ where $X_i = (X_{i1}, \ldots, X_{ij}, \ldots, X_{ip})^{\mathrm{T}} \in [0,1]^p$ and

$$Y_i = m(X_i) + \varepsilon_i \tag{2}$$

with $\varepsilon_i \sim N(0, \sigma^2)$ independent of X_i and

$$m(x) = \sum_{j=1}^{p} f_j(x_j). \tag{3}$$

Let μ denote the distribution of X, and let μ_j denote the marginal distribution of X_j for each $j = 1, \ldots, p$. For a function f_j on $[0, 1]$ denote its $L_2(\mu_j)$ norm by

$$\|f_j\|_{\mu_j} = \sqrt{\left\{ \int_0^1 f_j^2(x)\, \mathrm{d}\mu_j(x) \right\}} = \sqrt{\mathbb{E}\{f_j(X_j)^2\}}. \tag{4}$$

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 $j \in \{1, \ldots, p\}$, 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

$$\langle f_j, f_j' \rangle = \mathbb{E}\{f_j(X_j)\, f_j'(X_j)\} \tag{5}$$

and $\|f_j\| = \sqrt{\mathbb{E}\{f_j(X_j)^2\}} < \infty$. Let $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \ldots \oplus \mathcal{H}_p$ denote the Hilbert space of functions of $(x_1, \ldots, x_p)$ that have the additive form: $m(x) = \Sigma_j f_j(x_j)$, with $f_j \in \mathcal{H}_j$, $j = 1, \ldots, p$.

Let $\{\psi_{jk}, k = 0, 1, \ldots\}$ 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

$$\mathcal{T}_j = \left\{ f_j \in \mathcal{H}_j : f_j(x_j) = \sum_{k=0}^{\infty} \beta_{jk}\, \psi_{jk}(x_j),\ \sum_{k=0}^{\infty} \beta_{jk}^2 k^{2\nu_j} \leqslant C^2 \right\} \tag{6}$$

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 $v = (v_1, \ldots, v_k)^{\mathrm{T}}$ is a vector, we use the norms

$$\|v\| = \sqrt{\left(\sum_{j=1}^{k} v_j^2 \right)}, \qquad \|v\|_1 = \sum_{j=1}^{k} |v_j|, \qquad \|v\|_\infty = \max_j |v_j|. \tag{7}$$

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

$$\min_{f_j \in \mathcal{H}_j, 1 \leqslant j \leqslant p} \left[\mathbb{E}\left\{ Y - \sum_{j=1}^{p} f_j(X_j) \right\}^2 \right] \tag{8}$$

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:

$$\min_{\beta\in\mathbb{R}^p, g_j\in\mathcal{H}_j}\left[\mathbb{E}\left\{Y-\sum_{j=1}^{p}\beta_j g_j(X_j)\right\}^2\right] \tag{9}$$

subject to

$$\sum_{j=1}^{p}|\beta_j|\leqslant L, \tag{10}$$

$$\mathbb{E}(g_j^2)=1, \qquad j=1,\ldots,p, \tag{11}$$

noting that g_j is a function whereas $\beta=(\beta_1,\ldots,\beta_p)^{\mathrm{T}}$ is a vector. The constraint that β lies in the l_1-ball $\{\beta:\|\beta\|_1\leqslant L\}$ 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:

$$\mathcal{L}(f,\lambda)=\frac{1}{2}\mathbb{E}\left\{Y-\sum_{j=1}^{p}f_j(X_j)\right\}^2+\lambda\sum_{j=1}^{p}\sqrt{\mathbb{E}\{f_j^2(X_j)\}}. \tag{12}$$

Theorem 1. The minimizers $f_j\in\mathcal{H}_j$ of equation (12) satisfy

$$f_j=\left[1-\frac{\lambda}{\sqrt{\mathbb{E}(P_j^2)}}\right]_+P_j \qquad \text{almost surely} \tag{13}$$

where $[\cdot]_+$ denotes the positive part, and $P_j=\mathbb{E}(R_j|X_j)$ denotes the projection of the residual $R_j=Y-\Sigma_{k\neq j}f_k(X_k)$ 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_js 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:

$$\hat{P}_j=\mathcal{S}_jR_j \tag{14}$$

where $\mathcal{S}_j$ is a linear smoother, such as a local linear or kernel smoother. Let

$$\hat{s}_j=\frac{1}{\sqrt{n}}\|\hat{P}_j\|=\sqrt{\text{mean}(\hat{P}_j^2)} \tag{15}$$

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 $\hat{P}_j(i)=\hat{\beta}_jX_{ij}$. When the variables are standardized,

Table 1. SPAM backfitting algorithm†

Input: data (X_i, Y_i), regularization parameter λ
Initialize $\hat{f}_j = 0$, for $j = 1, \ldots, p$
Iterate until convergence, *for each* $j = 1, \ldots, p$

Step 1: compute the residual, $R_j = Y - \Sigma_{k \neq j} \hat{f}_k(X_k)$
Step 2: estimate $P_j = \mathbb{E}(R_j | X_j)$ by smoothing, $\hat{P}_j = \mathcal{S}_j R_j$
Step 3: estimate the norm, $\hat{s}_j^2 = (1/n)\Sigma_{i=1}^n \hat{P}_j^2(i)$
Step 4: soft threshold, $\hat{f}_j = [1 - \lambda/\hat{s}_j]_+ \hat{P}_j$
Step 5: centre, $\hat{f}_j \leftarrow \hat{f}_j -$ mean $(\hat{f}_j)$.

Output: component functions $\hat{f}_j$ and estimator $\hat{m}(X_i) = \Sigma_j \hat{f}_j(X_{ij})$

†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†

Input: data (X_i, Y_i), regularization parameter λ
Initialize $\hat{\beta}_j = 0$, for $j = 1, \ldots, p$
Iterate until convergence, *for each* $j = 1, \ldots, p$

Step 1: compute the residual, $R_j = Y - \Sigma_{k \neq j} \hat{\beta}_k X_k$
Step 2: project residual onto X_j, $P_j = X_j^{\mathrm{T}} R_j$
Step 3: soft threshold, $\hat{\beta}_j = [1 - \lambda/|P_j|]_+ P_j$

Output: estimator $\hat{m}(X_i) = \Sigma_j \hat{\beta}_j X_{ij}$

†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 $B_j = (\psi_{jk} : k = 1, 2, \ldots)$ is an orthonormal basis for $\mathcal{T}_j$ and that $\sup_x |\psi_{jk}(x)| \leqslant B$ for some B. Then

$$f_j(x_j) = \sum_{k=1}^{\infty} \beta_{jk}\, \psi_{jk}(x_j) \tag{16}$$

where $\beta_{jk} = \int f_j(x_j)\, \psi_{jk}(x_j)\, \mathrm{d}x_j$.

Let us also define

$$\tilde{f}_j(x_j) = \sum_{k=1}^{d} \beta_{jk}\, \psi_{jk}(x_j) \tag{17}$$

where $d = d_n$ is a truncation parameter. For the Sobolev space $\mathcal{T}_j$ 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 $\tilde{m} = \Sigma_j \tilde{f}_j$. 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 $\{\psi_{j1}, \ldots, \psi_{jd}\}$; 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 $\mathcal{S}_j = \Psi_j(\Psi_j^{\mathrm{T}}\Psi_j)^{-1}\Psi_j^{\mathrm{T}}$. In this case, the backfitting algorithm in Table 1 is a co-ordinate descent algorithm for minimizing

$$\frac{1}{2n}\left\| Y - \sum_{j=1}^{p} \Psi_j \beta_j \right\|_2^2 + \lambda \sum_{j=1}^{p} \sqrt{\left(\frac{1}{n}\beta_j^{\mathrm{T}}\Psi_j^{\mathrm{T}}\Psi_j\beta_j\right)}$$

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:

$$Y = \sum_{j=1}^{p_n} X_j \beta_j + \varepsilon = X\beta + \varepsilon, \tag{18}$$

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 jth 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 $X_j^{\mathrm{T}}X_j = I_{d_j}$, where I_{d_j} is the $d_j \times d_j$ identity matrix. We use $X = (X_1, \ldots, X_{p_n})$ to denote the full design matrix and use $\beta = (\beta_1^{\mathrm{T}}, \ldots, \beta_{p_n}^{\mathrm{T}})^{\mathrm{T}}$ to denote the parameter.

The *grouped lasso* estimator is defined as the solution of the following convex optimization problem:

$$\hat{\beta}(\lambda_n) = \arg\min_{\beta}\left(\|Y - X\beta\|_2^2 + \lambda_n \sum_{j=1}^{p_n} \sqrt{d_j}\|\beta_j\| \right) \tag{19}$$

where $\sqrt{d_j}$ scales the jth term to compensate for different group sizes.

It is obvious that, when $d_j = 1$ for $j = 1, \ldots, p_n$, the grouped lasso becomes the standard lasso. From the Karush–Kuhn–Tucker optimality conditions, a necessary and sufficient condition for $\hat{\beta} = (\hat{\beta}_1^{\mathrm{T}}, \ldots, \hat{\beta}_p^{\mathrm{T}})^{\mathrm{T}}$ to be the grouped lasso solution is

$$\begin{aligned} -X_j^{\mathrm{T}}(Y - X\hat{\beta}) + \frac{\lambda\sqrt{d_j}\hat{\beta}_j}{\|\hat{\beta}_j\|} &= \mathbf{0}, \qquad \forall \hat{\beta}_j \neq \mathbf{0}, \\ \|X_j^{\mathrm{T}}(Y - X\hat{\beta})\| &\leqslant \lambda\sqrt{d_j}, \qquad \forall \hat{\beta}_j = \mathbf{0}. \end{aligned} \tag{20}$$

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

$$\hat{\beta}_j = \left[1 - \frac{\lambda\sqrt{d_j}}{\|S_j\|}\right]_+ S_j \tag{21}$$

where $S_j = X_j^{\mathrm{T}}(Y - X\beta_{\setminus j})$, with $\beta_{\setminus j} = (\beta_1^{\mathrm{T}}, \ldots, \beta_{j-1}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \beta_{j+1}^{\mathrm{T}}, \ldots, \beta_{p_n}^{\mathrm{T}})$. 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 $\Sigma_j|\beta_j|$ 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 ($\Sigma_j|\beta_j| \leqslant L$). 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 jth variable, i.e. $\nu_j = \text{tr}(\mathcal{S}_j)$ where $\mathcal{S}_j$ is the smoothing matrix for the jth dimension. Also let $\hat{\sigma}^2$ be an estimate of the variance. Define the total effective degrees of freedom as

$$\text{df}(\lambda) = \sum_j \nu_j I(\|\hat{f}_j\| \neq 0). \tag{22}$$

Two estimates of risk are

$$C_p = \frac{1}{n}\sum_{i=1}^{n}\left\{Y_i - \sum_{j=1}^{p}\hat{f}_j(X_j)\right\}^2 + \frac{2\hat{\sigma}^2}{n}\text{df}(\lambda) \tag{23}$$

and

$$\text{GCV}(\lambda) = \frac{(1/n)\sum_{i=1}^{n}\left\{Y_i - \sum_j \hat{f}_j(X_{ij})\right\}^2}{\{1 - \text{df}(\lambda)/n\}^2}. \tag{24}$$

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.

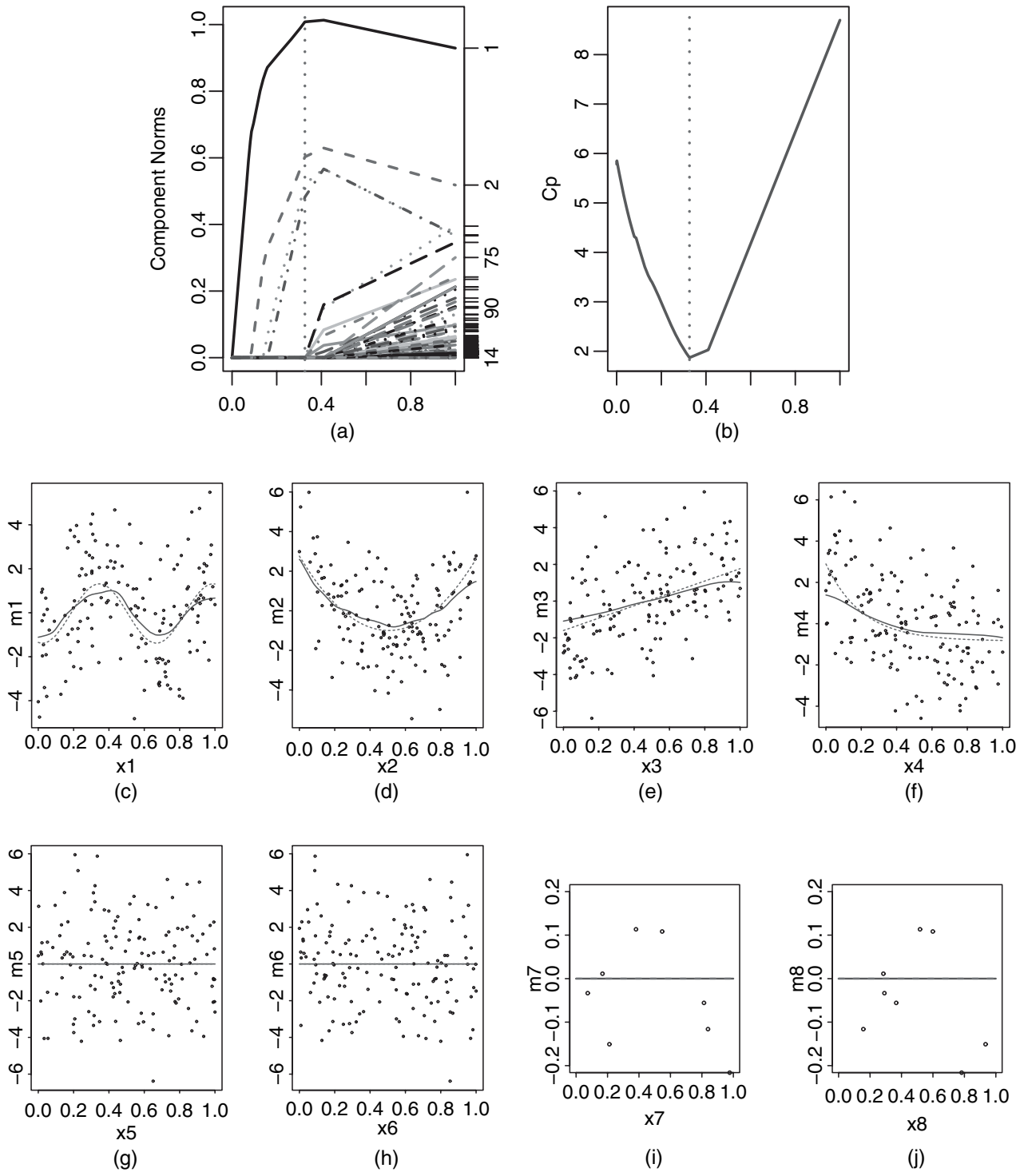


Fig. 1. Simulated data: (a) empirical l_2-norm of the estimated components as plotted against the regularization parameter λ (the value on the x-axis is proportional to $\Sigma_j \|\hat{f}_j\|$); (b) C_p-scores against the amount of regularization (⋮, value of λ which has the smallest C_p-score); estimated (———) *versus* true additive component functions (– – –) for (c)–(f) the first four relevant dimensions and (g)–(j) the first four irrelevant dimensions ((c) $l_1 = 97.05$; (d) $l_1 = 88.36$; (e) $l_1 = 90.65$; (f) $l_1 = 79.26$; (g)–(j) $l_1 = 0$)

$$\frac{1}{n}\sum_{i=1}^{n} f_j(X_{ij}) = 0,$$
$$\frac{1}{n-1}\sum_{i=1}^{n} f_j^2(X_{ij}) = 1. \tag{25}$$

The results of applying SPAMs are summarized in Fig. 1, using the plug-in bandwidths

$$h_j = 0.6\,\mathrm{sd}(X_j)/n^{1/5}.$$

Fig. 1(a) shows regularization paths as the parameter λ varies; each curve is a plot of $\|\hat{f}_j(\lambda)\|$ *versus*

$$\sum_{k=1}^{p} \|\hat{f}_k(\lambda)\| \Big/ \max_\lambda \left\{ \sum_{k=1}^{p} \|\hat{f}_k(\lambda)\| \right\} \tag{26}$$

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.

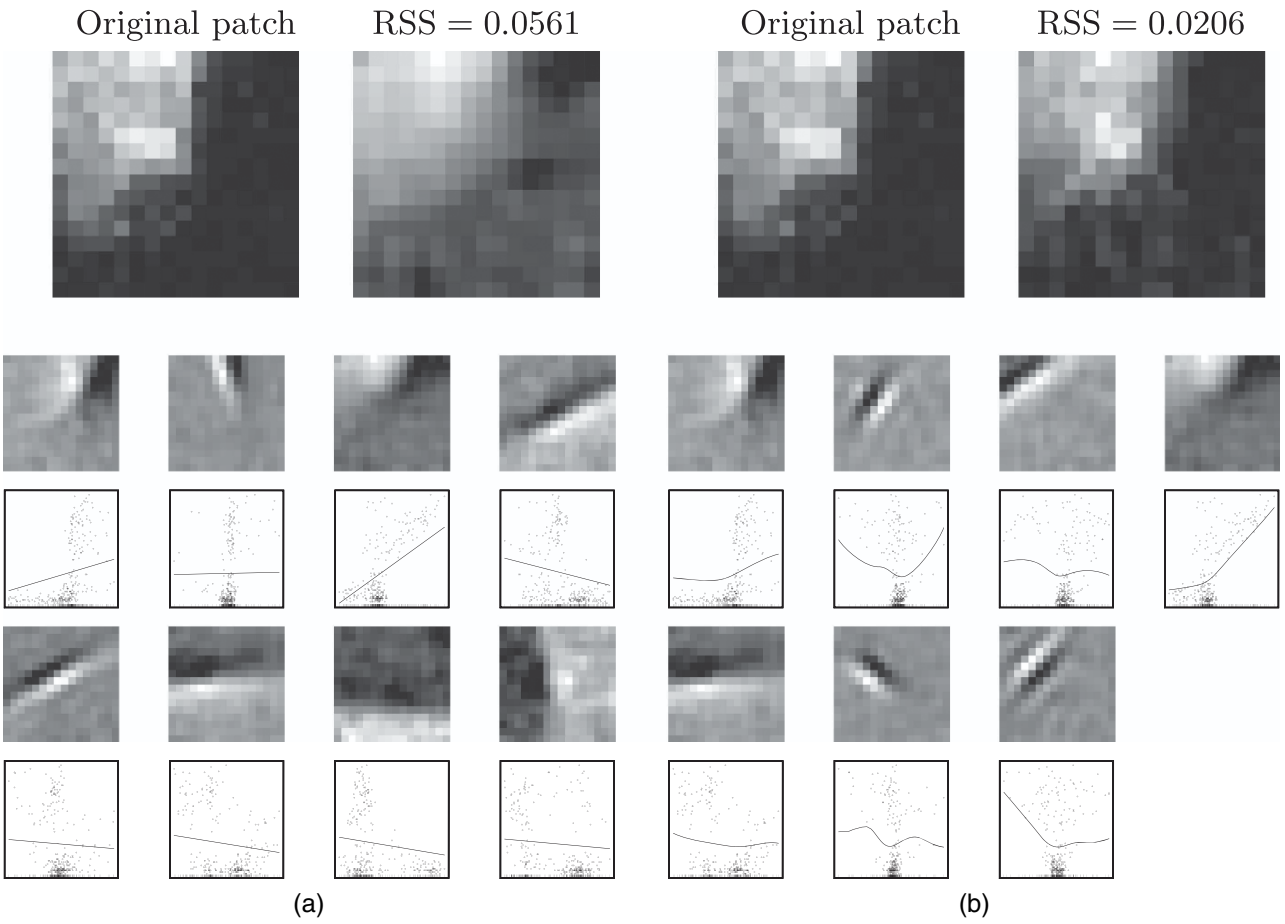


Fig. 2. Comparison of sparse reconstructions by using (a) the lasso and (b) SPAMs

Let $\{y^i\}_{i=1,\ldots,N}$ 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

$$\min_{\beta, X} \left\{ \sum_{i=1}^{N} \left(\frac{1}{2n} \|y^i - X\beta^i\|^2 + \lambda \|\beta^i\|_1 \right) \right\} \tag{27}$$

such that

$$\|X_j\| \leqslant 1. \tag{28}$$

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:

$$\min_{f, X} \left[\sum_{i=1}^{N} \left\{ \frac{1}{2n} \left\| y^i - \sum_{j=1}^{p} f_j^i(X_j) \right\|^2 + \lambda \sum_{j=1}^{p} \|f_j^i\| \right\} \right] \tag{29}$$

such that

$$\|X_j\| \leqslant 1, \qquad j = 1, \ldots, p. \tag{30}$$

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 $f_j(X_j) = \beta_j^{*\mathrm{T}} X_j$, several researchers have shown that, under certain conditions on n and p, the number of relevant variables $s = |\mathrm{supp}(\beta^*)|$, and the design matrix X, the lasso recovers the sparsity pattern asymptotically, i.e. the lasso estimator $\hat{\beta}_n$ is *sparsistent*:

$$\mathbb{P}\{\mathrm{supp}(\beta^*) = \mathrm{supp}(\hat{\beta}_n)\} \to 1. \tag{31}$$

Here, $\mathrm{supp}(\beta) = \{j : \beta_j \neq 0\}$. 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

$$Y_i = \sum_{j=1}^{p} \sum_{k=1}^{\infty} \beta^*_{jk}\, \psi_{jk}(X_{ij}) + \varepsilon_i. \tag{32}$$

To simplify the notation, let β_j be the d_n-dimensional vector $\{\beta_{jk}, k = 1, \ldots, d_n\}$ and let Ψ_j be the $n \times d_n$ matrix $\Psi_j(i,k) = \psi_{jk}(X_{ij})$. If $A \subset \{1, \ldots, p\}$, 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 $\{\psi_{j1}, \ldots, \psi_{jd}\}$. Our optimization problem in this setting is

$$\min_{\beta} \left\{ \frac{1}{2n} \left\| Y - \sum_{j=1}^{p} \Psi_j \beta_j \right\|_2^2 + \lambda \sum_{j=1}^{p} \sqrt{\left(\frac{1}{n} \beta_j^{\mathrm{T}} \Psi_j^{\mathrm{T}} \Psi_j \beta_j \right)} \right\}. \tag{33}$$

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

$$\min_{\beta} \left\{ \frac{1}{2n} \| R_j - \Psi_j \beta_j \|_2^2 + \lambda_n \sqrt{\left(\frac{1}{n} \beta_j^{\mathrm{T}} \Psi_j^{\mathrm{T}} \Psi_j \beta_j \right)} \right\}$$

where $\|v\|_2^2$ denotes $\Sigma_{i=1}^{n} v_i^2$ and $R_j = Y - \Sigma_{l \neq j} \Psi_l \beta_l$ is the residual for f_j. The sparse backfitting algorithm thus solves

$$\min_{\beta} \{ R_n(\beta) + \lambda_n\, \Omega(\beta) \} = \min_{\beta} \left(\frac{1}{2n} \left\| Y - \sum_{j=1}^{p} \Psi_j \beta_j \right\|_2^2 + \lambda_n \sum_{j=1}^{p} \left\| \frac{1}{\sqrt{n}} \Psi_j \beta_j \right\|_2 \right) \tag{34}$$

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 $\hat{f}_j(x_j) = \Sigma_{k=1}^{d_n} \hat{\beta}_{jk}\, \psi_{jk}(x_j)$. 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:

$$\Lambda_{\max} \left(\frac{1}{n} \Psi_S^{\mathrm{T}} \Psi_S \right) \leqslant C_{\max} < \infty, \tag{35}$$

$$\Lambda_{\min} \left(\frac{1}{n} \Psi_S^{\mathrm{T}} \Psi_S \right) \geqslant C_{\min} > 0, \tag{36}$$

$$\max_{j \in S^{\mathrm{c}}} \left\| \left(\frac{1}{n} \Psi_j^{\mathrm{T}} \Psi_S \right) \left(\frac{1}{n} \Psi_S^{\mathrm{T}} \Psi_S \right)^{-1} \right\| \leqslant \sqrt{\left(\frac{C_{\min}}{C_{\max}} \right)} \frac{1 - \delta}{\sqrt{s}}, \qquad \text{for some } 0 < \delta \leqslant 1. \tag{37}$$

Assume that the truncation dimension d_n satisfies $d_n \to \infty$ 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:

$$\frac{s}{d_n \lambda_n} \to 0, \tag{38}$$

$$\frac{d_n \log\{d_n(p-s)\}}{n\lambda_n^2} \to 0, \tag{39}$$

$$\frac{1}{\rho_n^*}\left[\sqrt{\left\{\frac{\log(sd_n)}{n}\right\}} + \frac{s^{3/2}}{d_n} + \lambda_n\sqrt{(sd_n)}\right] \to 0 \tag{40}$$

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 $\Psi_j^{\mathrm{T}}\Psi$ 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

$$\|\Psi_{S^c}^{\mathrm{T}}\Psi_S(\Psi_S^{\mathrm{T}}\Psi_S)^{-1}\|_\infty = \max_{j\in S^c}\|\Psi_j^{\mathrm{T}}\Psi_S(\Psi_S^{\mathrm{T}}\Psi_S)^{-1}\|_\infty \tag{41}$$

$$\leqslant \sqrt{\left(\frac{C_{\min}d_n}{C_{\max}}\right)}(1-\delta) \tag{42}$$

since $(1/\sqrt{n})\|A\|_\infty \leqslant \|A\| \leqslant \sqrt{m}\|A\|_\infty$ 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

$$d_n \asymp n^{1/3}, \tag{43}$$

$$\lambda_n \asymp \frac{\log(np)}{n^{1/3}}, \tag{44}$$

$$\frac{1}{\rho_n^*} = o\left\{\frac{n^{1/6}}{\log(np)}\right\} \tag{45}$$

then $\mathbb{P}(\hat{S}_n = S) \to 1$.

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 $f(x) = \Sigma_k \beta_k \psi_k(x)$ is in the Sobolev space of order $\nu > \frac{1}{2}$, so that $\Sigma_{i=1}^\infty \beta_i^2 i^{2\nu} \leqslant C^2$ for some constant C. Then

$$\|f\|_2 = \|\beta\|_2 \leqslant c\|\beta\|_\infty^{2\nu/(2\nu+1)} \tag{46}$$

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 Greenshtein 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

$$R(v) = \mathbb{E}\{Y - v(X)\}^2. \tag{47}$$

When $v(x) = \Sigma_j \beta_j g_j(x_j)$ we also write the risk as $R(\beta, g)$ where $\beta = (\beta_1, \ldots, \beta_p)$ and $g = (g_1, \ldots, g_p)$. Following Greenshtein 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

$$R(\hat{m}_n) - R(m_n^*) \xrightarrow{\mathrm{P}} 0 \tag{48}$$

where

$$m_n^* = \arg\min_{v \in \mathcal{M}_n} \{R(v)\} \tag{49}$$

is the predictive oracle. Greenshtein and Ritov (2004) showed that the lasso is persistent for $\mathcal{M}_n = \{l(x) = x^{\mathrm{T}}\beta : \|\beta\|_1 \leqslant L_n\}$ 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

$$\frac{1}{n}\sum_{i=1}^{n}\left\{Y_i - \sum_j \beta_j g_j(X_{ij})\right\}^2 \tag{50}$$

subject to $\|\beta\|_1 \leqslant L_n$ 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

$$\mathcal{M}_n = \left\{m : m(x) = \sum_{j=1}^{p_n} \beta_j g_j(x_j) : \mathbb{E}(g_j) = 0, \mathbb{E}(g_j^2) = 1, \sum_j |\beta_j| \leqslant L_n\right\} \tag{51}$$

and let $m_n^* = \arg\min_{v \in \mathcal{M}_n}\{R(v)\}$.

Theorem 3. Suppose that $p_n \leqslant \exp(n^\xi)$ for some $\xi < 1$. Then,

$$R(\hat{m}_n) - R(m_n^*) = O_P\left(\frac{L_n^2}{n^{(1-\xi)/2}}\right) \tag{52}$$

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

$$\min_{\{f_j \in \mathcal{H}_j\}} \{\mathcal{L}(f, \lambda)\} \equiv \frac{1}{2} \mathbb{E}\left\{ Y - \sum_{j=1}^{p} f_j(X_j) \right\}^2 + \lambda \sum_{j=1}^{p} \sqrt{\mathbb{E}\{f_j(X_j)^2\}} \tag{53}$$

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

$$\partial_j \mathcal{L}(f, \lambda; \eta_j) = \tfrac{1}{2} \mathbb{E}\{(f_j - R_j + \lambda v_j)\eta_j\} = 0 \tag{54}$$

where $R_j = Y - \Sigma_{k \neq j} f_k$ 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 $v_j \in \{u_j \in \mathcal{H}_j | \mathbb{E}(u_j^2) \leqslant 1\}$ otherwise.

Using iterated expectations, the above condition can be rewritten as

$$\mathbb{E}[\{f_j + \lambda v_j - \mathbb{E}(R_j | X_j)\}\eta_j] = 0. \tag{55}$$

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

$$\mathbb{E}[\{f_j(x_j) - \mathbb{E}(R_j | X_j = x_j) + \lambda v_j(x_j)\}^2] = 0, \tag{56}$$

i.e.

$$f_j + \lambda v_j = \mathbb{E}(R_j | X_j) \qquad \text{almost everywhere.} \tag{57}$$

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

$$\sqrt{\mathbb{E}(P_j^2)} = \sqrt{\mathbb{E}[\{f_j + \lambda f_j / \sqrt{\mathbb{E}(f_j^2)}\}^2]} \tag{58}$$

$$= \left\{ 1 + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}} \right\} \sqrt{\mathbb{E}(f_j^2)} \tag{59}$$

$$= \sqrt{\mathbb{E}(f_j^2)} + \lambda \tag{60}$$

$$\geqslant \lambda. \tag{61}$$

If $\mathbb{E}(f_j^2) = 0$, then $f_j = 0$ almost everywhere, and $\sqrt{\mathbb{E}(v_j^2)} \leqslant 1$. Equation (57) then implies that

$$\sqrt{\mathbb{E}(P_j^2)} \leqslant \lambda. \tag{62}$$

We thus obtain the equivalence

$$\sqrt{\mathbb{E}(P_j^2)} \leqslant \lambda \Leftrightarrow f_j = 0 \qquad \text{almost everywhere.} \tag{63}$$

Rewriting equation (57) in light of result (63), we obtain

$$\left\{1 + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}}\right\} f_j = P_j \qquad \text{if } \sqrt{\mathbb{E}(P_j^2)} > \lambda,$$
$$f_j = 0 \qquad \text{otherwise.}$$

Using equation (60), we thus arrive at the soft thresholding update for f_j:

$$f_j = \left[1 - \frac{\lambda}{\sqrt{\mathbb{E}(P_j^2)}}\right]_+ P_j \tag{64}$$

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

$$\frac{1}{n}\Psi^{\mathrm{T}}\left(\sum_j \Psi_j \hat{\beta}_j - Y\right) + \lambda_n \hat{g} = 0. \tag{65}$$

The subdifferential $\partial\Omega(\beta)$ is the set of vectors $g \in \mathbb{R}^{pd_n}$ satisfying

$$g_j = \frac{(1/n)\Psi_j^{\mathrm{T}}\Psi_j\beta_j}{\sqrt{\{(1/n)\beta_j^{\mathrm{T}}\Psi_j^{\mathrm{T}}\Psi_j\beta_j\}}} \qquad \text{if } \beta_j \neq 0,$$
$$g_j^{\mathrm{T}}\left(\frac{1}{n}\Psi_j^{\mathrm{T}}\Psi_j\right)^{-1} g_j \leqslant 1 \qquad \text{if } \beta_j = 0.$$

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 $\mathrm{supp}(\hat{\beta}) = \mathrm{supp}(\beta^*)$ 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 $\mathrm{supp}(\beta^*)$, 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 $\hat{\beta}_{S^{\mathrm{c}}} = 0$.
(b) Set $\hat{g}_S = \partial\Omega(\beta^*)_S$.
(c) With these settings of $\hat{\beta}_{S^{\mathrm{c}}}$ and $\hat{g}_S$, obtain $\hat{\beta}_S$ and $\hat{g}_{S^{\mathrm{c}}}$ 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

$$\hat{\beta}_j \neq 0 \qquad \text{for } j \in S, \tag{66a}$$

$$g_j^{\mathrm{T}}\left(\frac{1}{n}\Psi_j^{\mathrm{T}}\Psi_j\right)^{-1} g_j < 1 \qquad \text{for } j \in S^{\mathrm{c}}. \tag{66b}$$

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

$$\hat{g}_j = \frac{(1/n)\Psi_j^{\mathrm{T}}\Psi_j\beta_j^*}{\sqrt{\{(1/n)\beta_j^{*\mathrm{T}}\Psi_j^{\mathrm{T}}\Psi_j\beta_j^*\}}} \qquad \text{for } j \in S,$$

the stationarity condition for $\hat{\beta}_S$ is given by

$$\frac{1}{n}\Psi_S^{\mathrm{T}}(\Psi_S\hat{\beta}_S - Y) + \lambda_n\hat{g}_S = 0. \tag{67}$$

Let $V = Y - \Psi_S\beta_S^* - W$ denote the error due to finite truncation of the orthogonal basis, where $W = (\varepsilon_1, \ldots, \varepsilon_n)^{\mathrm{T}}$. Then the stationarity condition (67) can be simplified as

$$\frac{1}{n}\Psi_S^{\mathrm{T}}\Psi_S(\hat{\beta}_S - \beta_S^*) - \frac{1}{n}\Psi_S^{\mathrm{T}}W - \frac{1}{n}\Psi_S^{\mathrm{T}}V + \lambda_n\hat{g}_S = 0,$$

so that

$$\hat{\beta}_S - \beta_S^* = \left(\frac{1}{n}\Psi_S^{\mathrm{T}}\Psi_S\right)^{-1}\left(\frac{1}{n}\Psi_S^{\mathrm{T}}W + \frac{1}{n}\Psi_S^{\mathrm{T}}V - \lambda_n\hat{g}_S\right), \tag{68}$$

where we have used the assumption that $(1/n)\Psi_S^{\mathrm{T}}\Psi_S$ 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

$$\operatorname{supp}(\beta_S^*) = \operatorname{supp}(\hat{\beta}_S) = \{j : \|\hat{\beta}_j\|_\infty \neq 0\},$$

so that condition (66a) would be satisfied. Using $\Sigma_{SS} = (1/n)(\Psi_S^{\mathrm{T}}\Psi_S)$ to simplify the notation, we have the l_∞-bound,

$$\|\hat{\beta}_S - \beta_S^*\|_\infty \leqslant \underbrace{\left\|\Sigma_{SS}^{-1}\left(\frac{1}{n}\Psi_S^{\mathrm{T}}W\right)\right\|_\infty}_{T_1} + \underbrace{\left\|\Sigma_{SS}^{-1}\left(\frac{1}{n}\Psi_S^{\mathrm{T}}V\right)\right\|_\infty}_{T_2} + \lambda_n \underbrace{\|\Sigma_{SS}^{-1}\hat{g}_S\|_\infty}_{T_3}. \tag{69}$$

We now proceed to bound the quantities T_1, T_2 and T_3.

A.2.2. Bounding T_3
Note that, for $j \in S$,

$$1 = g_j^{\mathrm{T}}\left(\frac{1}{n}\Psi_j^{\mathrm{T}}\Psi_j\right)^{-1}g_j \geqslant \frac{1}{C_{\max}}\|g_j\|^2,$$

and thus $|g_j\| \leqslant \sqrt{C_{\max}}$. Noting further that

$$\|g_S\|_\infty = \max_{j\in S}(\|g_j\|_\infty) \leqslant \max_{j\in S}(\|g_j\|_2) \leqslant \sqrt{C_{\max}}, \tag{70}$$

it follows that

$$T_3 := \|\Sigma_{SS}^{-1}\hat{g}_S\|_\infty \leqslant \sqrt{C_{\max}}\|\Sigma_{SS}^{-1}\|_\infty. \tag{71}$$

A.2.3. Bounding T_2
We proceed in two steps; we first bound $\|V\|_\infty$ and use this to bound $\|(1/n)\Psi_S^{\mathrm{T}}V\|_\infty$. Note that, as we are working over the Sobolev spaces $\mathcal{S}_j$ of order 2,

$$\begin{aligned}
|V_i| &= \left|\sum_{j\in S}\sum_{k=d_n+1}^{\infty}\beta_{jk}^*\Psi_{jk}(X_{ij})\right| \leqslant B\sum_{j\in S}\sum_{k=d_n+1}^{\infty}|\beta_{jk}^*| \\
&= B\sum_{j\in S}\sum_{k=d_n+1}^{\infty}\frac{|\beta_{jk}^*|k^2}{k^2} \leqslant B\sum_{j\in S}\sqrt{\left(\sum_{k=d_n+1}^{\infty}\beta_{jk}^{*2}k^4\right)}\sqrt{\left(\sum_{k=d_n+1}^{\infty}\frac{1}{k^4}\right)} \\
&\leqslant sBC\sqrt{\left(\sum_{k=d_n+1}^{\infty}\frac{1}{k^4}\right)} \leqslant \frac{sB'}{d_n^{3/2}},
\end{aligned}$$

for some constant $B' > 0$. It follows that

$$\left| \frac{1}{n} \Psi_{jk}^{\mathrm{T}} V \right| \leqslant \left| \frac{1}{n} \sum_i \Psi_{jk}(X_{ij}) \right| \|V\|_\infty \leqslant \frac{Ds}{d_n^{3/2}}, \tag{72}$$

where D denotes a generic constant. Thus,

$$T_2 := \left\| \Sigma_{SS}^{-1} \left(\frac{1}{n} \Psi_S^{\mathrm{T}} V \right) \right\|_\infty \leqslant \|\Sigma_{SS}^{-1}\|_\infty \frac{Ds}{d_n^{3/2}}. \tag{73}$$

A.2.4. Bounding T_1

Let $Z = T_1 = \Sigma_{SS}^{-1}(1/n)\Psi_S^{\mathrm{T}} W$. Note that $W \sim N(0, \sigma^2 I)$, so that Z is Gaussian as well, with mean 0. Consider its lth component, $Z_l = e_l^{\mathrm{T}} Z$. Then $\mathbb{E}(Z_l) = 0$, and

$$\mathrm{var}(Z_l) = \frac{\sigma^2}{n} e_l^{\mathrm{T}} \Sigma_{SS}^{-1} e_l \leqslant \frac{\sigma^2}{C_{\min} n}.$$

By Gaussian comparison results (Ledoux and Talagrand, 1991), we have then that

$$\mathbb{E}(\|Z\|_\infty) \leqslant 3\sqrt{\{\log(sd_n)\|\mathrm{var}(Z)\|_\infty\}} \leqslant 3\sigma \sqrt{\left\{ \frac{\log(sd_n)}{nC_{\min}} \right\}}. \tag{74}$$

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

$$\begin{aligned}
\mathbb{P}\left(\|\hat{\beta}_S - \beta_S^*\|_\infty > \frac{\rho_n^*}{2} \right) &\leqslant \mathbb{P}\left\{ \|Z\|_\infty + \|\Sigma_{SS}^{-1}\|_\infty (Dsd_n^{-3/2} + \lambda_n \sqrt{C_{\max}}) > \frac{\rho_n^*}{2} \right\} \\
&\leqslant \frac{2}{\rho_n^*} \{ \mathbb{E}(\|Z\|_\infty) + \|\Sigma_{SS}^{-1}\|_\infty (Dsd_n^{-3/2} + \lambda_n \sqrt{C_{\max}}) \} \\
&\leqslant \frac{2}{\rho_n^*} \left[3\sigma \sqrt{\left\{ \frac{\log(sd_n)}{nC_{\min}} \right\}} + \|\Sigma_{SS}^{-1}\|_\infty \left(\frac{Ds}{d_n^{3/2}} + \lambda_n \sqrt{C_{\max}} \right) \right],
\end{aligned}$$

which converges to 0 under the condition that

$$\frac{1}{\rho_n^*} \left[\sqrt{\left\{ \frac{\log(sd_n)}{n} \right\}} + \left\| \left(\frac{1}{n} \Psi_S^{\mathrm{T}} \Psi_S \right)^{-1} \right\|_\infty \left(\frac{s}{d_n^{3/2}} + \lambda_n \right) \right] \to 0. \tag{75}$$

Noting that

$$\left\| \left(\frac{1}{n} \Psi_S^{\mathrm{T}} \Psi_S \right)^{-1} \right\|_\infty \leqslant \frac{\sqrt{(sd_n)}}{C_{\min}}, \tag{76}$$

it follows that condition (75) holds when

$$\frac{1}{\rho_n^*} \left[\sqrt{\left\{ \frac{\log(sd_n)}{n} \right\}} + \frac{s^{3/2}}{d_n} + \lambda_n \sqrt{(sd_n)} \right] \to 0. \tag{77}$$

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

$$\frac{1}{n} \Psi_j^{\mathrm{T}} (\Psi_S \hat{\beta}_S - \Psi_S \beta_S^* - W - V) + \lambda_n \hat{g}_j = 0.$$

It then follows from equation (68) that

$$\begin{aligned}\hat{g}_{S^c} &= \frac{1}{\lambda_n}\left\{\frac{1}{n}\Psi_{S^c}^{\mathrm{T}}\Psi_S(\beta_S^* - \hat{\beta}_S) + \frac{1}{n}\Psi_{S^c}^{\mathrm{T}}(W+V)\right\} \\ &= \frac{1}{\lambda_n}\left\{\frac{1}{n}\Psi_{S^c}^{\mathrm{T}}\Psi_S\left(\frac{1}{n}\Psi_S^{\mathrm{T}}\Psi_S\right)^{-1}\left(\lambda_n\hat{g}_S - \frac{1}{n}\Psi_S^{\mathrm{T}}W - \frac{1}{n}\Psi_S^{\mathrm{T}}V\right) + \frac{1}{n}\Psi_{S^c}^{\mathrm{T}}(W+V)\right\},\end{aligned}$$

so

$$\hat{g}_{S^c} = \frac{1}{\lambda_n}\left\{\Sigma_{S^cS}\Sigma_{SS}^{-1}\left(\lambda_n\hat{g}_S - \frac{1}{n}\Psi_S^{\mathrm{T}}W - \frac{1}{n}\Psi_S^{\mathrm{T}}V\right) + \frac{1}{n}\Psi_{S^c}^{\mathrm{T}}(W+V)\right\}. \tag{78}$$

Condition (66b) requires that

$$g_j^{\mathrm{T}}\left(\frac{1}{n}\Psi_j^{\mathrm{T}}\Psi_j\right)^{-1} g_j < 1, \tag{79}$$

for all $j \in S^c$. Since

$$g_j^{\mathrm{T}}\left(\frac{1}{n}\Psi_j^{\mathrm{T}}\Psi_j\right)^{-1} g_j \leqslant \frac{1}{C_{\min}}\|g_j\|^2 \tag{80}$$

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

$$\mu_j = \mathbb{E}(\hat{g}_j) = \Sigma_{jS}\Sigma_{SS}^{-1}\left(\hat{g}_S - \frac{1}{\lambda_n}\frac{1}{n}\Psi_S^{\mathrm{T}}V\right) - \frac{1}{\lambda_n}\frac{1}{n}\Psi_j^{\mathrm{T}}V.$$

This can be bounded as

$$\begin{aligned}\|\mu_j\| &\leqslant \|\Sigma_{jS}\Sigma_{SS}^{-1}\|\left(\|\hat{g}_S\| + \frac{1}{\lambda_n}\left\|\frac{1}{n}\Psi_S^{\mathrm{T}}V\right\|\right) + \frac{1}{\lambda_n}\left\|\frac{1}{n}\Psi_j^{\mathrm{T}}V\right\| \\ &= \|\Sigma_{jS}\Sigma_{SS}^{-1}\|\left\{\sqrt{(sC_{\max})} + \frac{1}{\lambda_n}\left\|\frac{1}{n}\Psi_S^{\mathrm{T}}V\right\|\right\} + \frac{1}{\lambda_n}\left\|\frac{1}{n}\Psi_j^{\mathrm{T}}V\right\|.\end{aligned} \tag{81}$$

Using the bound $\|\Psi_j^{\mathrm{T}}V\|_\infty \leqslant Ds/d_n^{3/2}$ from equation (72), we have

$$\left\|\frac{1}{n}\Psi_j^{\mathrm{T}}V\right\| \leqslant \sqrt{d_n}\left\|\frac{1}{n}\Psi_j^{\mathrm{T}}V\right\|_\infty \leqslant \frac{Ds}{d_n},$$

and hence

$$\left\|\frac{1}{n}\Psi_S^{\mathrm{T}}V\right\| \leqslant \sqrt{s}\left\|\frac{1}{n}\Psi_S^{\mathrm{T}}V\right\|_\infty \leqslant \frac{Ds^{3/2}}{d_n}.$$

Substituting in the bound (81) on the mean μ_j,

$$\|\mu_j\| \leqslant \|\Sigma_{jS}\Sigma_{SS}^{-1}\|\left\{\sqrt{(sC_{\max})} + \frac{Ds^{3/2}}{\lambda_n d_n}\right\} + \frac{Ds}{\lambda_n d_n}. \tag{82}$$

Assumptions (37) and (38) of the theorem can be rewritten as

$$\|\Sigma_{jS}\Sigma_{SS}^{-1}\| \leqslant \sqrt{\left(\frac{C_{\min}}{C_{\max}}\right)}\frac{1-\delta}{\sqrt{s}} \qquad \text{for some } \delta > 0, \tag{83}$$

$$\frac{s}{\lambda_n d_n} \to 0. \tag{84}$$

Thus the bound on the mean becomes

$$\|\mu_j\| \leqslant \sqrt{C_{\min}}(1-\delta) + \frac{2Ds}{\lambda_n d_n} < \sqrt{C_{\min}},$$

for sufficiently large n. It therefore suffices, for condition (66b) to be satisfied, to show that

$$\mathbb{P}\left(\max_{j \in S^c} \|\hat{g}_j - \mu_j\|_\infty > \frac{\delta}{2\sqrt{d_n}} \right) \to 0, \tag{85}$$

since this implies that

$$\begin{aligned} \|\hat{g}_j\| &\leqslant \|\mu_j\| + \|\hat{g}_j - \mu_j\| \\ &\leqslant \|\mu_j\| + \sqrt{d_n}\|\hat{g}_j - \mu_j\|_\infty \\ &\leqslant \sqrt{C_{\min}}(1-\delta) + \frac{\delta}{2} + o(1), \end{aligned}$$

with probability approaching 1. To show result (85), we again appeal to Gaussian comparison results. Define

$$Z_j = \Psi_j^{\mathrm{T}}(I - \Psi_S(\Psi_S^{\mathrm{T}}\Psi_S)^{-1}\Psi_S^{\mathrm{T}})\frac{W}{n}, \tag{86}$$

for $j \in S^c$. Then Z_j are zero-mean Gaussian random variables, and we need to show that

$$\mathbb{P}\left\{ \max_{j \in S^c}\left(\frac{\|Z_j\|_\infty}{\lambda_n} \right) \geqslant \frac{\delta}{2\sqrt{d_n}} \right\} \to \infty. \tag{87}$$

A calculation shows that $\mathbb{E}(Z_{jk}^2) \leqslant \sigma^2/n$. Therefore, we have by Markov's inequality and Gaussian comparison that

$$\begin{aligned} \mathbb{P}\left\{ \max_{j \in S^c}\left(\frac{\|Z_j\|_\infty}{\lambda_n} \right) \geqslant \frac{\delta}{2\sqrt{d_n}} \right\} &\leqslant \frac{2\sqrt{d_n}}{\delta\lambda_n}\mathbb{E}(\max_{jk}|Z_{jk}|) \\ &\leqslant \frac{2\sqrt{d_n}}{\delta\lambda_n}[3\sqrt{\log\{(p-s)d_n\}}\max_{jk}\{\sqrt{\mathbb{E}(Z_{jk}^2)}\}] \\ &\leqslant \frac{6\sigma}{\delta\lambda_n}\sqrt{\left[\frac{d_n\log\{(p-s)d_n\}}{n}\right]}, \end{aligned}$$

which converges to 0 given the assumption (39) of the theorem that

$$\frac{\lambda_n^2 n}{d_n\log\{(p-s)d_n\}} \to \infty.$$

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

$$\|f\|_2^2 = \sum_{i=1}^{\infty}\beta_i^2 \tag{88}$$

$$\leqslant \|\beta\|_\infty \sum_{i=1}^{\infty}|\beta_i| \tag{89}$$

$$= \|\beta\|_\infty \sum_{i=1}^{k}|\beta_i| + \|\beta\|_\infty \sum_{i=k+1}^{\infty}|\beta_i| \tag{90}$$

$$\leqslant k\|\beta\|_\infty^2 + \|\beta\|_\infty \sum_{i=k+1}^{\infty}\frac{i^\nu|\beta_i|}{i^\nu} \tag{91}$$

$$\leqslant k\|\beta\|_\infty^2 + \|\beta\|_\infty \sqrt{\left(\sum_{i=1}^{\infty} \beta_i^2 i^{2\nu}\right)} \sqrt{\left(\sum_{i=k+1}^{\infty} \frac{1}{i^{2\nu}}\right)} \tag{92}$$

$$\leqslant k\|\beta\|_\infty^2 + \|\beta\|_\infty C \sqrt{\left(\frac{k^{1-2\nu}}{2\nu-1}\right)}, \tag{93}$$

where the last inequality uses the bound

$$\sum_{i=k+1}^{\infty} i^{-2\nu} \leqslant \int_k^\infty x^{-2\nu} \mathrm{d}x = \frac{k^{1-2\nu}}{2\nu-1}. \tag{94}$$

Let k^* be the index that minimizes expression (93). Some calculus shows that k^* satisfies

$$c_1 \|\beta\|_\infty^{-2/(2\nu+1)} \leqslant k^* \leqslant c_2 \|\beta\|_\infty^{-2/(2\nu+1)} \tag{95}$$

for some constants c_1 and c_2. Using the above expression in expression (93) then yields

$$\|f\|_2^2 \leqslant \|\beta\|_\infty (c_2 \|\beta\|_\infty^{(2\nu-1)/(2\nu+1)} + c_1' \|\beta\|_\infty^{(2\nu-1)/(2\nu+1)}) \tag{96}$$

$$= c\|\beta\|_\infty^{4\nu/(2\nu+1)} \tag{97}$$

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 $B = \{(l_1, u_1), \ldots, (l_k, u_k)\}$ such that $\|u_j - l_j\|_\infty \leqslant \varepsilon$, $1 \leqslant j \leqslant k$, and such that for every $m \in \mathcal{M}$ there exists $(l, u) \in B$ such that $l \leqslant m \leqslant u$. For the Sobolev space $\mathcal{T}_j$,

$$\log\{N_{[]}(\varepsilon, \mathcal{T}_j)\} \leqslant K \left(\frac{1}{\varepsilon}\right)^{1/2} \tag{98}$$

for some $K > 0$; see van der Vaart (1998). The bracketing integral is defined to be

$$J_{[]}(\delta, \mathcal{M}) = \int_0^\delta \sqrt{\log\{N_{[]}(u, \mathcal{M})\}}\, \mathrm{d}u. \tag{99}$$

From corollary 19.35 of van der Vaart (1998),

$$\mathbb{E}\left\{ \sup_{g \in \mathcal{M}} |\hat{\mu}(g) - \mu(g)| \right\} \leqslant \frac{C\, J_{[]}(\|F\|_\infty, \mathcal{M})}{\sqrt{n}} \tag{100}$$

for some $C > 0$, where $F(x) = \sup_{g \in \mathcal{M}} |g(x)|$, $\mu(g) = \mathbb{E}\{g(X)\}$ and $\hat{\mu}(g) = n^{-1} \Sigma_{i=1}^n g(X_i)$.

Set $Z \equiv (Z_0, \ldots, Z_p) = (Y, X_1, \ldots, X_p)$ and note that

$$R(\beta, g) = \sum_{j=0}^{p} \sum_{k=0}^{p} \beta_j \beta_k\, \mathbb{E}\{g_j(Z_j)\, g_k(Z_k)\} \tag{101}$$

where we define $g_0(z_0) = z_0$ and $\beta_0 = -1$. Also define

$$\hat{R}(\beta, g) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=0}^{p} \sum_{k=0}^{p} \beta_j \beta_k\, g_j(Z_{ij})\, g_k(Z_{ik}). \tag{102}$$

Hence $\hat{m}_n$ is the minimizer of $\hat{R}(\beta, g)$ subject to the constraint $\Sigma_j \beta_j\, g_j(x_j) \in \mathcal{M}_n(L_n)$ and $g_j \in \mathcal{T}_j$. For all (β, g),

$$|\hat{R}(\beta, g) - R(\beta, g)| \leqslant \|\beta\|_1^2 \max_{jk} \sup_{g_j \in \mathcal{S}_j, g_k \in \mathcal{S}_k} |\hat{\mu}_{jk}(g) - \mu_{jk}(g)| \tag{103}$$

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

$$\log\{N_{[]}(\varepsilon, \mathcal{M}_n)\} \leqslant 2\log(p_n) + K\left(\frac{1}{\varepsilon}\right)^{1/2}. \tag{104}$$

Hence, $J_{[]}(C, \mathcal{M}_n) = O\{\sqrt{\log(p_n)}\}$ and it follows from inequality (100) and Markov's inequality that

$$\max_{jk} \sup_{g_j \in \mathcal{S}_j, g_k \in \mathcal{S}_k} |\hat{\mu}_{jk}(g) - \mu_{jk}(g)| = O_P\left[\sqrt{\left\{\frac{\log(p_n)}{n}\right\}}\right] = O_P\left(\frac{1}{n^{(1-\xi)/2}}\right). \tag{105}$$

We conclude that

$$\sup_{g \in \mathcal{M}} |\hat{R}(g) - R(g)| = O_P\left(\frac{L_n^2}{n^{(1-\xi)/2}}\right). \tag{106}$$

Therefore,

$$\begin{aligned} R(m^*) &\leqslant R(\hat{m}_n) \leqslant \hat{R}(\hat{m}_n) + O_P\left(\frac{L_n^2}{n^{(1-\xi)/2}}\right) \\ &\leqslant \hat{R}(m^*) + O_P\left(\frac{L_n^2}{n^{(1-\xi)/2}}\right) \leqslant R(m^*) + O_P\left(\frac{L_n^2}{n^{(1-\xi)/2}}\right) \end{aligned}$$

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 $(X_1, Y_1), \ldots, (X_n, Y_n) \sim P$ where $X_i \in \mathbb{R}^d$ and $Y_i \in \{0, 1, \ldots, K-1\}$. 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 $X_1, \ldots, X_{256}$ 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 $h : \mathcal{X} \to \{0, \ldots, K-1\}$ 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 $\big\{x \in \mathbb{R}^d \ : \ H(x) = 0\big\}$, 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

$$R(h) = \mathbb{P}\big(Y \neq h(X)\big) \tag{1}$$

and the *empirical classification error* or *training error* is

$$\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} I\big(h(X_i) \neq Y_i\big). \tag{2}$$

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, $h : \mathcal{X} \to \{0, 1\}$
H	linear discriminant function, $H(x) = \beta_0 + \beta^T x$ and $h(x) = I\big(H(x) > 0\big)$
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.

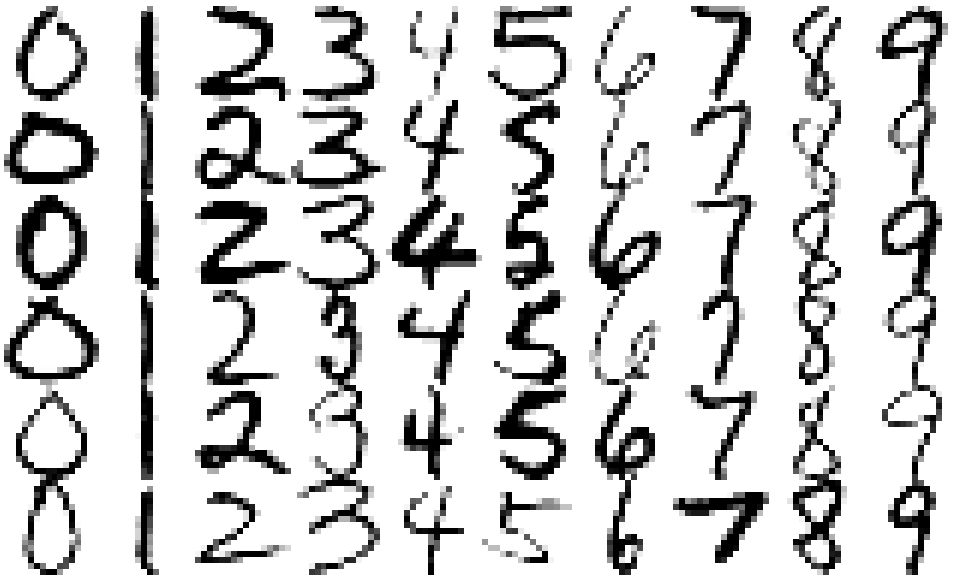


Figure 2: Examples from the zipcode data.

Theorem 1 *The rule h that minimizes $R(h)$ is*

$$h^*(x) = \begin{cases} 1 & \text{if } m(x) > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

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

$$R(h) = \mathbb{P}\big(\{Y \neq h(X)\}\big) = \int \mathbb{P}\big(Y \neq h(X)|X = x\big) dP_X(x).$$

It suffices to show that

$$\mathbb{P}\big(Y \neq h(X)|X = x\big) - \mathbb{P}\big(Y \neq h^*(X)|X = x\big) \geq 0 \quad \text{for all } x \in \mathcal{X}. \tag{4}$$

Now,

$$\begin{aligned}
\mathbb{P}\big(Y \neq h(X)|X = x\big) &= 1 - \mathbb{P}\big(Y = h(X)|X = x\big) && (5)\\
&= 1 - \Big(\mathbb{P}\big(Y = 1, h(X) = 1|X = x\big) + \mathbb{P}\big(Y = 0, h(X) = 0|X = x\big)\Big) && (6)\\
&= 1 - \Big(h(x)\mathbb{P}(Y = 1|X = x) + \big(1 - h(x)\big)\mathbb{P}(Y = 0|X = x)\Big) && (7)\\
&= 1 - \Big(h(x)m(x) + \big(1 - h(x)\big)\big(1 - m(x)\big)\Big). && (8)
\end{aligned}$$

Hence,

$$\begin{aligned}
&\mathbb{P}\big(Y \neq h(X)|X=x\big) - \mathbb{P}\big(Y \neq h^*(X)|X=x\big) \\
&= \Big(h^*(x)m(x) + \big(1-h^*(x)\big)\big(1-m(x)\big)\Big) - \Big(h(x)m(x) + \big(1-h(x)\big)\big(1-m(x)\big)\Big) \\
&= \big(2m(x)-1\big)\big(h^*(x)-h(x)\big) = 2\left(m(x)-\frac{1}{2}\right)\big(h^*(x)-h(x)\big). \qquad (9)
\end{aligned}$$

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). □

We can rewrite h^* in a different way. From Bayes' theorem

$$\begin{aligned}
m(x) &= \mathbb{P}(Y=1|X=x) = \frac{p(x|Y=1)\mathbb{P}(Y=1)}{p(x|Y=1)\mathbb{P}(Y=1) + p(x|Y=0)\mathbb{P}(Y=0)} \\
&= \frac{\pi_1 p_1(x)}{\pi_1 p_1(x) + (1-\pi_1)p_0(x)}. \qquad (10)
\end{aligned}$$

where $\pi_1 = \mathbb{P}(Y=1)$. From the above equality, we have that

$$m(x) > \frac{1}{2} \quad \text{is equivalent to} \quad \frac{p_1(x)}{p_0(x)} > \frac{1-\pi_1}{\pi_1}. \qquad (11)$$

Thus the Bayes rule can be rewritten as

$$h^*(x) = \begin{cases} 1 & \text{if } \frac{p_1(x)}{p_0(x)} > \frac{1-\pi_1}{\pi_1} \\ 0 & \text{otherwise.} \end{cases} \qquad (12)$$

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,

$$R(h) - R^* = \underbrace{R(h) - R(h_o)}_{\text{distance from oracle}} + \underbrace{R(h_o) - R^*}_{\text{distance of oracle from Bayes error}}. \qquad (13)$$

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 $\widehat{m}(x)$ be an estimate of $m^*(x)$ and define the *plug-in classification rule*:

$$\widehat{h}(x) = \begin{cases} 1 & \text{if } \widehat{m}(x) > \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

We have the following theorem.

Theorem 2 *The risk of the plug-in classifier rule in* (14) *satisfies*

$$R(\widehat{h}) - R^* \leq 2\sqrt{\int (\widehat{m}(x) - m^*(x))^2 dP_X(x)}.$$

Proof. In the proof of Theorem 1 we showed that

$$\begin{aligned} \mathbb{P}\big(Y \neq \widehat{h}(X)|X = x\big) - \mathbb{P}\big(Y \neq h^*(X)|X = x\big) &= \big(2\widehat{m}(x) - 1\big)\big(h^*(x) - \widehat{h}(x)\big) \\ &= \big|2\widehat{m}(x) - 1\big| I\big(h^*(x) \neq \widehat{h}(x)\big) = 2\big|\widehat{m}(x) - 1/2\big| I\big(h^*(x) \neq \widehat{h}(x)\big). \end{aligned}$$

Now, when $h^*(x) \neq \widehat{h}(x)$, there are two possible cases: (i) $\widehat{h}(x) = 1$ and $h^*(x) = 0$; (ii) $\widehat{h}(x) = 0$ and $h^*(x) = 1$. In both cases, we have that $\big|\widehat{m}(x) - m^*(x)\big| \geq \big|\widehat{m}(x) - 1/2\big|$. Therefore,

$$\begin{aligned} \mathbb{P}\big(\widehat{h}(X) \neq Y\big) - \mathbb{P}\big(h^*(X) \neq Y\big) &= 2\int \big|\widehat{m}(x) - 1/2\big| I\big(h^*(x) \neq \widehat{h}(x)\big) dP_X(x) \\ &\leq 2\int \big|\widehat{m}(x) - m^*(x)\big| I\big(h^*(x) \neq \widehat{h}(x)\big) dP_X(x) \\ &\leq 2\int \big|\widehat{m}(x) - m^*(x)\big| dP_X(x) \qquad (15) \\ &\leq 2\sqrt{\int \big(\widehat{m}(x) - m^*(x)\big)^2 dP_X(x)}. \qquad (16) \end{aligned}$$

The last inequality follows from the fact that $\mathbb{E}|Z| \leq \sqrt{\mathbb{E}Z^2}$ for any Z. □

This theorem implies that if the regression estimate $\widehat{m}(x)$ 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 $\widehat{m}$ to be far from $m^*(x)$ and still lead to a good classifier. As long as $\widehat{m}(x)$ 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.*

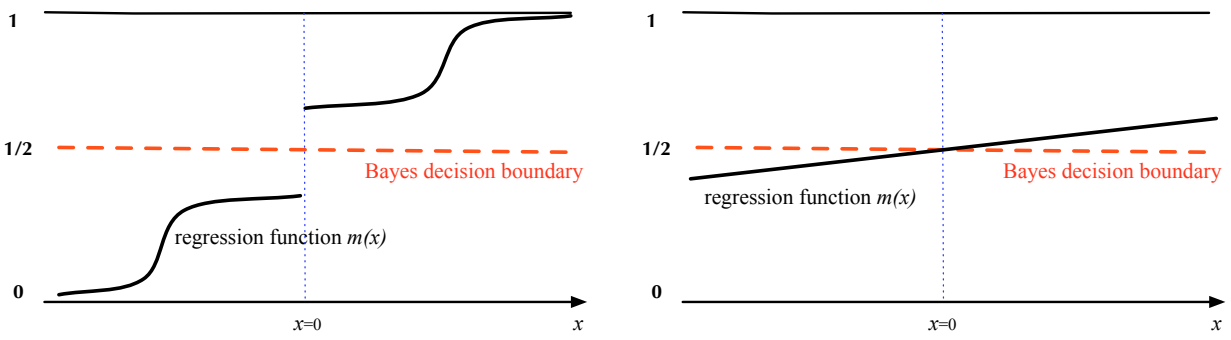


Figure 3: The Bayes rule is $h^*(x) = I(x > 0)$ in both plots, which show the regression function $m(x) = \mathbb{E}(Y|x)$ for two problems. The left plot shows an easy problem; there is little ambiguity around the decision boundary. The right plot shows a hard problem; it is hard to know from the data 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 $R(\widehat{m}) \to 0$. We have that

$$\begin{aligned}
R(\widehat{h}) - R(h_*) &\leq 2 \int |\widehat{m}(x) - m_*(x)| I(h_*(x) \neq \widehat{h}(x)) dP(x) \\
&= 2 \int |\widehat{m}(x) - m_*(x)| I(h_*(x) \neq \widehat{h}(x)) I(m_*(x) \neq 1/2) dP(x) \\
&= 2\mathbb{E}\left[|\widehat{m}(X) - m_*(X)| I(h_*(X) \neq \widehat{h}(X)) I(m_*(X) \neq 1/2) \right] \\
&\leq 2\mathbb{E}\left[|\widehat{m}(X) - m_*(X)| I(h_*(X) \neq \widehat{h}(X)) I(|m_*(X) - 1/2| \leq \epsilon, m_*(X) \neq 1/2) \right] \\
&\quad + 2\mathbb{E}\left[|\widehat{m}(X) - m_*(X)| I(h_*(X) \neq \widehat{h}(X)) I(|m_*(X) - 1/2| > \epsilon) \right] \\
&\leq 2\sqrt{R(\widehat{m})}(a^{1/2} + b^{1/2})
\end{aligned}$$

where

$$a = P(h_*(X) \neq \widehat{h}(X), |m_*(X) - 1/2| \leq \epsilon, m_*(X) \neq 1/2)$$

and

$$b = P(h_*(X) \neq \widehat{h}(X), |m_*(X) - 1/2| > \epsilon).$$

Now

$$b \leq P(|\widehat{m}(X) - m_*(X)| > \epsilon) \leq \frac{R(\widehat{m})}{\epsilon^2} \to 0$$

so

$$\lim_{n \to \infty} \frac{R(\widehat{h}) - R(h_*)}{\sqrt{R(\widehat{m})}} \leq 2a^{1/2}.$$

But $a \to 0$ as $\epsilon \to 0$ so So

$$\frac{R(\widehat{h}) - R(h_*)}{\sqrt{R(\widehat{m})}} \to 0.$$

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

$$\inf_{\widehat{m}} \sup_{m\in\mathcal{M}} R(\widehat{h}) \asymp r_n^2 \quad \text{and} \quad \inf_{\widehat{h}} \sup_{m\in\mathcal{M}} [R(\widehat{h}) - R(h_*)] \asymp r_n$$

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 $\widehat{m}$ satifies an exponential inequality then $\frac{R(\widehat{h})-R(h_*)}{\sqrt{R(\widehat{m})}}$ 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 $\widehat{\beta}$ to be the value of β that minimizes

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

The problem with this approach is that it is difficult to minimize $\widehat{R}_n(\beta)$. The theory for the ERM is straightforward. First, let us recall the following. Let $\mathcal{H}$ be a set of clasifiers and let $h_* = \text{argmin}_{h\in\mathcal{H}} R(h)$. Let $\widehat{h}$ minimize the empirical risk. If $\sup_{h\in\mathcal{H}} |\widehat{R}(h) - R(h)| \leq \epsilon$ then $R(\widehat{h}) \leq R(h_*) + 2\epsilon$. To see this, note that if if $\sup_{h\in\mathcal{H}} |\widehat{R}(h) - R(h)| \leq \epsilon$ then, using the fact that $\widehat{h}$ minimizes $\widehat{R}$,

$$R(h_*) \leq R(\widehat{h}) \leq \widehat{R}(\widehat{h}) + \epsilon \leq \widehat{R}(h_*) + \epsilon \leq R(h_*) + 2\epsilon.$$

So we need to bound

$$P(\sup_{h\in\mathcal{H}} |\widehat{R}(h) - R(h)| \leq \epsilon).$$

If $\mathcal{H}$ has finite VC dimension r then

$$P(\sup_{h\in\mathcal{H}} |\widehat{R}(h) - R(h)| \leq \epsilon) \leq 8(n+1)^r e^{-n\epsilon^2/32}.$$

Now hald-spaces have VC dimension $r = d + 1$. So

$$P(\sup_{h\in\mathcal{H}} |\widehat{R}(h) - R(h)| \leq \epsilon) \leq 8(n+1)^{d+1} e^{-n\epsilon^2/32}.$$

We conclude that $P(R(\widehat{h}) - R(h_*) > 2\epsilon) \leq 8n^{d+1} e^{-n\epsilon^2/32}$.

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 ourselved 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*

$$h^*(x) = \begin{cases} 1 & \text{if } r_1^2 < r_0^2 + 2\log\left(\frac{\pi_1}{1-\pi_1}\right) + \log\left(\frac{|\Sigma_0|}{|\Sigma_1|}\right) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

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 $h^*(x) = I\big(\pi_1 p_1(x) > (1 - \pi_1)p_0(x)\big)$. Plug-in the specific forms of p_0 and p_1 and take the logarithms we get $h^*(x) = 1$ if and only if

$$\begin{aligned}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1) - 2\log\pi_1 + \log\big(|\Sigma_1|\big) \\ < (x - \mu_0)^T \Sigma_0^{-1}(x - \mu_0) - 2\log(1 - \pi_1) + \log\big(|\Sigma_0|\big).\end{aligned} \tag{18}$$

The theorem immediately follows from some simple algebra. □

Let $\pi_0 = 1 - \pi_1$. An equivalent way of expressing the Bayes rule is

$$h^*(x) = \text{argmax}_{k \in \{0,1\}} \delta_k(x) \tag{19}$$

where

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \log \pi_k \tag{20}$$

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

$$\widehat{\pi}_0 = \frac{1}{n}\sum_{i=1}^{n}(1 - Y_i), \quad \widehat{\pi}_1 = \frac{1}{n}\sum_{i=1}^{n} Y_i, \tag{21}$$

$$\widehat{\mu}_0 = \frac{1}{n_0}\sum_{i:\ Y_i=0} X_i, \quad \widehat{\mu}_1 = \frac{1}{n_1}\sum_{i:\ Y_i=1} X_i, \tag{22}$$

$$\widehat{\Sigma}_0 = \frac{1}{n_0 - 1}\sum_{i:\ Y_i=0} (X_i - \widehat{\mu}_0)(X_i - \widehat{\mu}_0)^T, \tag{23}$$

$$\widehat{\Sigma}_1 = \frac{1}{n_1 - 1}\sum_{i:\ Y_i=1} (X_i - \widehat{\mu}_1)(X_i - \widehat{\mu}_1)^T, \tag{24}$$

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

$$h^*(x) = \text{argmax}_k \delta_k(x) \tag{25}$$

where now

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + \log \pi_k. \tag{26}$$

Hence, the classifier is linear. The parameters are estimated as before, except that we use a pooled estimate of the Σ:

$$\widehat{\Sigma} = \frac{(n_0 - 1)\widehat{\Sigma}_0 + (n_1 - 1)\widehat{\Sigma}_1}{n_0 + n_1 - 2}. \tag{27}$$

The classification rule is

$$h^*(x) = \begin{cases} 1 & \text{if } \delta_1(x) > \delta_0(x) \\ 0 & \text{otherwise.} \end{cases} \tag{28}$$

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, $Y \in \{0, \ldots, K-1\}$ 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*

$$h^*(x) = \text{argmax}_k \mathbb{P}(Y = k | X = x) \tag{29}$$

Proof. We have

$$\begin{aligned} R(h) &= 1 - \mathbb{P}(h(X) = Y) && (30)\\ &= 1 - \sum_{k=0}^{K-1} \mathbb{P}(h(X) = k, Y = k) && (31)\\ &= 1 - \sum_{k=0}^{K-1} \mathbb{E}\Big[I\big(h(X) = k\big) \mathbb{P}(Y = k | X) \Big] && (32) \end{aligned}$$

It's clear that $h^*(X) = \text{argmax}_k \mathbb{P}(Y = k|X)$ achieves the minimized classification error $1 - \mathbb{E}\big[\max_k \mathbb{P}(Y = k|X)\big]$. □

Let $\pi_k = \mathbb{P}(Y = k)$. The next theorem extends QDA and LDA to the multiclass setting.

Theorem 6 *Suppose that $Y \in \{0, \ldots, K-1\}$ with $K \geq 2$. If $p_k(x) = p(x|Y = k)$ is Gaussian : $X|Y = k \sim N\big(\mu_k, \Sigma_k\big)$, the Bayes rule for the multiclass QDA can be written as*

$$h^*(x) = \text{argmax}_k \delta_k(x)$$

where

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log \pi_k. \tag{33}$$

If all Gaussians have an equal variance Σ, then

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k. \tag{34}$$

Let $n_k = \sum_i I(y_i = k)$ for $k = 0, \ldots, K-1$. The estimated sample quantities of π_k, μ_k, Σ_k, and Σ are:

$$\widehat{\pi}_k = \frac{1}{n}\sum_{i=1}^{n} I(y_i = k), \quad \widehat{\mu}_k = \frac{1}{n_k}\sum_{i:\ Y_i=k} X_i, \tag{35}$$

$$\widehat{\Sigma}_k = \frac{1}{n_k - 1}\sum_{i:\ Y_i=k} (X_i - \widehat{\mu}_k)(X_i - \widehat{\mu}_k)^T, \tag{36}$$

$$\widehat{\Sigma} = \frac{\sum_{k=0}^{K-1}(n_k - 1)\widehat{\Sigma}_k}{n - K}. \tag{37}$$

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.*

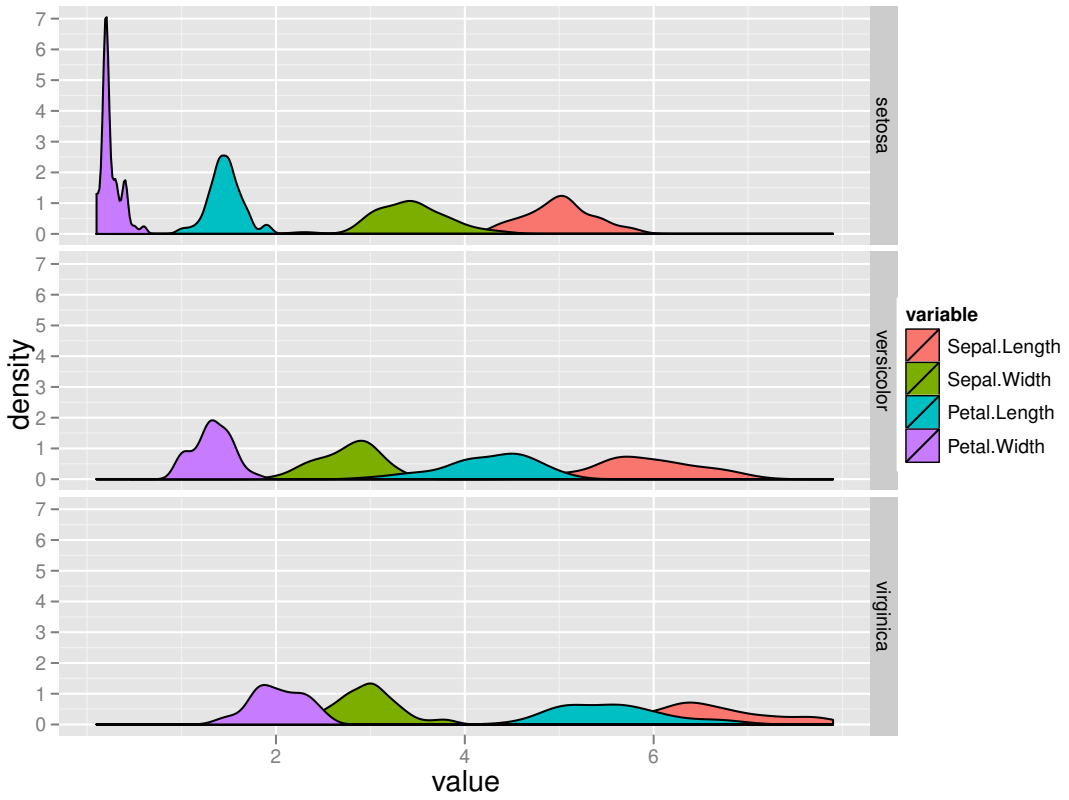


Figure 4: The Iris data: The estimated densities for different features are plotted within each class. It's easy to see that the distributions of petal length and petal width are quite different across different classes, which suggests that they are very informative features.

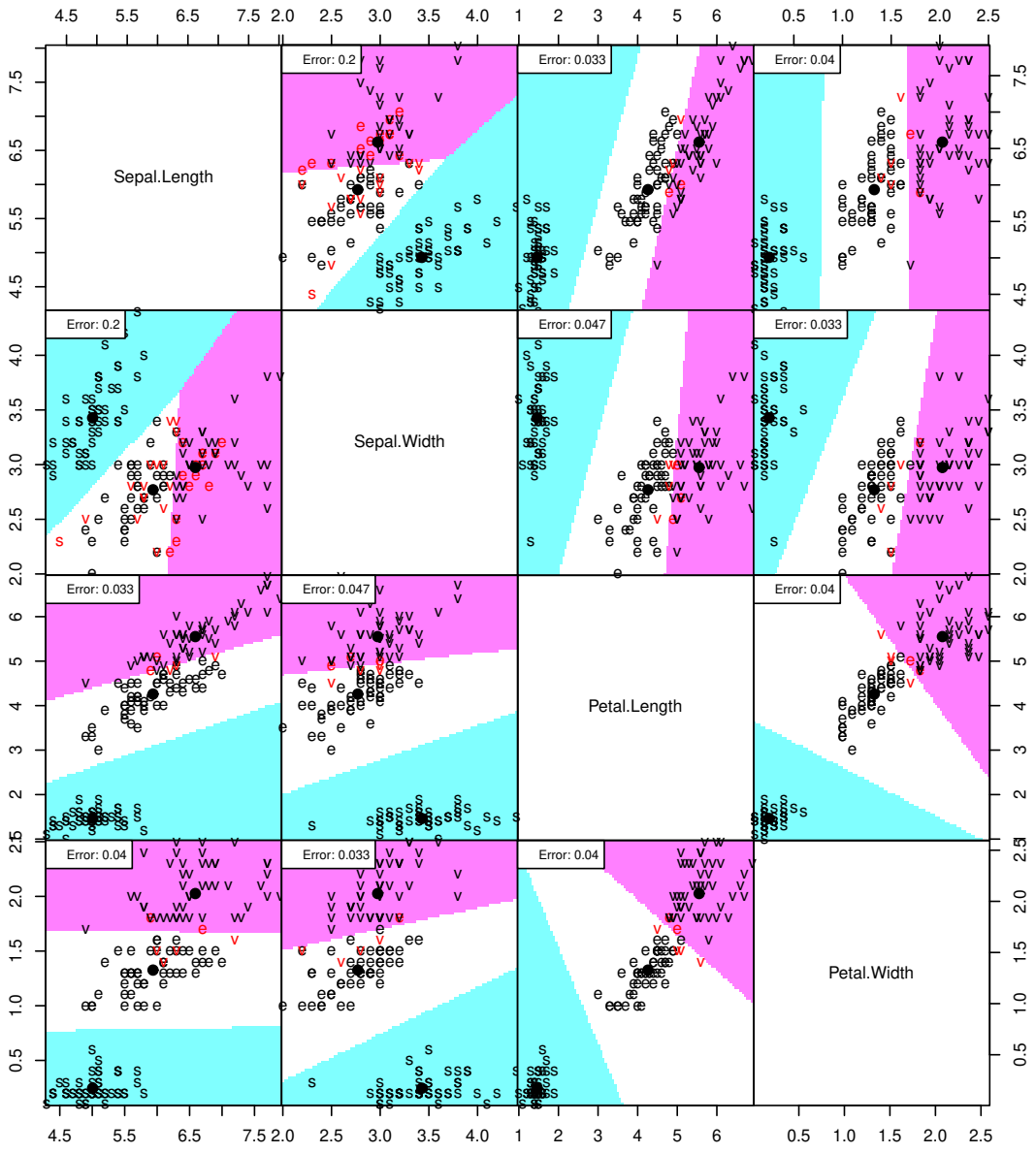


Figure 5: Classifying the Iris data using LDA. The multiple figure array illustrates the classification of observations based on LDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to a given pair of features before fitting the 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.

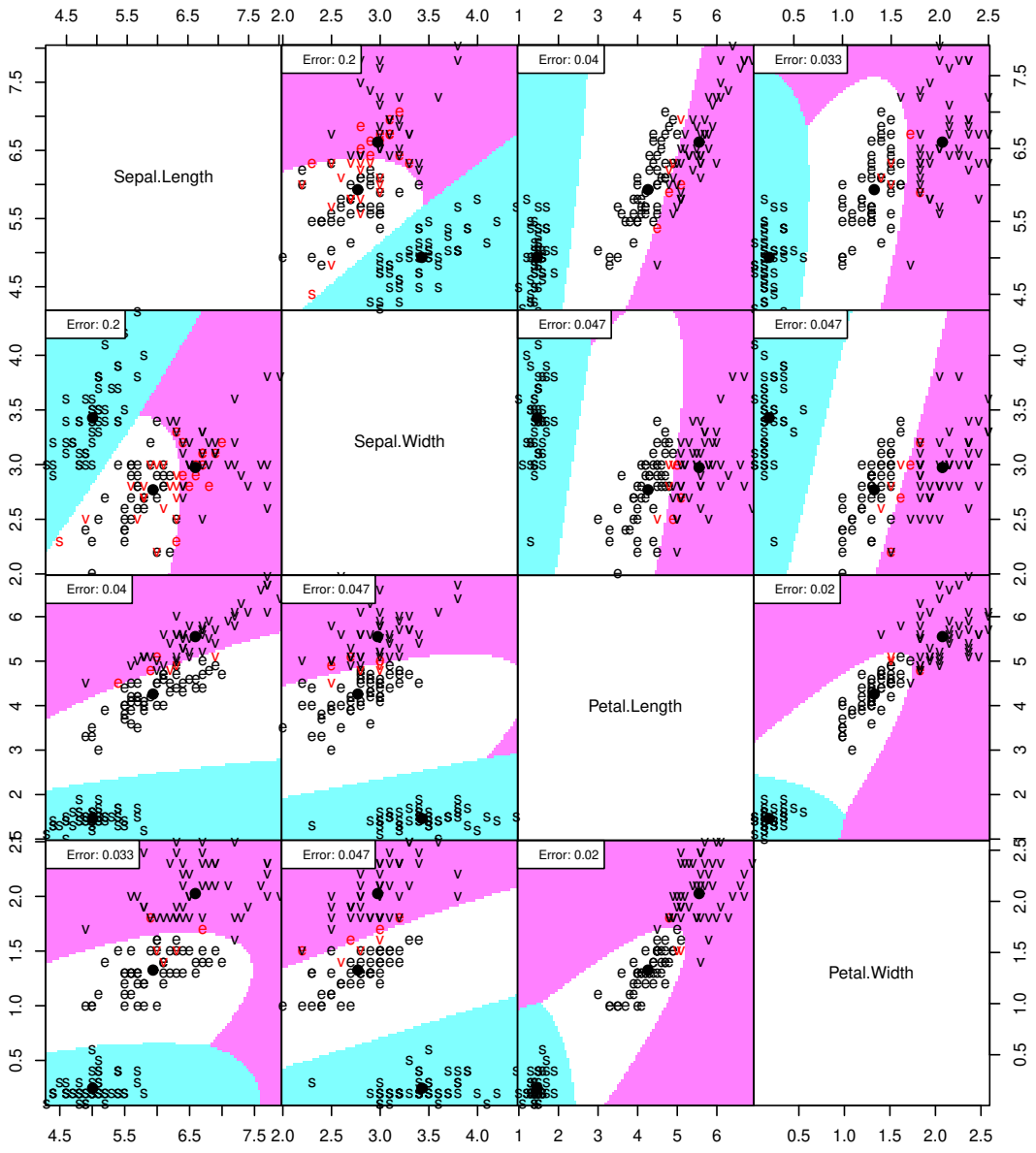


Figure 6: Classifying the Iris data using QDA. The multiple figure array illustrates the classification of observations based on QDA for every combination of two features. The classification boundaries are displayed and the classification error by simply casting the data onto these two features are calculated. 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 $X = (X_1, \ldots, X_d)^T$ with a linear combination $U = w^T X = \sum_{j=1}^d w_j X_j$. The goal is to choose the vector $w = (w_1, \ldots, w_d)^T$ 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

$$\begin{aligned} J(w) &= \frac{\big(\mathbb{E}(U|Y=0) - \mathbb{E}(U|Y=1)\big)^2}{w^T \Sigma w} \\ &= \frac{(w^T \mu_0 - w^T \mu_1)^2}{w^T \Sigma w} \\ &= \frac{w^T (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T \Sigma w}. \end{aligned}$$

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 $\widehat{\mu}_j$ 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

$$\widehat{J}(w) = \frac{w^T S_B w}{w^T S_W w} \tag{38}$$

where

$$\begin{aligned} S_B &= (\widehat{\mu}_0 - \widehat{\mu}_1)(\widehat{\mu}_0 - \widehat{\mu}_1)^T, \\ S_W &= \frac{(n_0 - 1)S_0 + (n_1 - 1)S_1}{(n_0 - 1) + (n_1 - 1)}. \end{aligned}$$

Theorem 8 *The vector*

$$\widehat{w} = S_W^{-1}(\widehat{\mu}_0 - \widehat{\mu}_1) \tag{39}$$

is a maximizer of $\widehat{J}(w)$.

Proof. Maximizing $\widehat{J}(w)$ 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 $\widehat{w}$ should be the eigenvector of $S_W^{-1}S_B$ corresponding to the largest eigenvalue. The key observation is that $S_B = (\widehat{\mu}_0 - \widehat{\mu}_1)(\widehat{\mu}_0 - \widehat{\mu}_1)^T$, which implies that for any vector w, $S_B w$ must be in the direction of $\widehat{\mu}_0 - \widehat{\mu}_1$. The desired result immediately follows. □

We call

$$f(x) = \widehat{w}^T x = (\widehat{\mu}_0 - \widehat{\mu}_1)^T S_W^{-1} x \tag{40}$$

the *Fisher linear discriminant function*. Given a cutting threshold $c_m \in \mathbb{R}$, Fisher's classification rule is

$$h(x) = \begin{cases} 0 & \text{if } \widehat{w}^T x \geq c_m \\ 1 & \text{if } \widehat{w}^T x < c_m. \end{cases} \tag{41}$$

Fisher's rule is the same as the Gaussian LDA rule in (26) when

$$c_m = \frac{1}{2}(\widehat{\mu}_0 - \widehat{\mu}_1)^T S_W^{-1}(\widehat{\mu}_0 + \widehat{\mu}_1) - \log\left(\frac{\widehat{\pi}_0}{\widehat{\pi}_1}\right). \tag{42}$$

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 $\widehat{m}(x)$, use the classification rule

$$\widehat{h}(x) = \begin{cases} 1 & \text{if } \widehat{m}(x) > \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{43}$$

For binary classification problems, one possible choice is the linear regression model

$$Y = m(X) + \epsilon = \beta_0 + \sum_{j=1}^{d} \beta_j X_j + \epsilon. \tag{44}$$

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 $Y_1, \ldots, Y_n$ is

$$\mathcal{L}(\pi_1) = \prod_{i=1}^{n} p(Y_i; \pi_1) = \prod_{i=1}^{n} \pi_1^{Y_i}(1 - \pi_1)^{1-Y_i}. \tag{45}$$

In the logistic regression model, we assume that

$$m(x) = \mathbb{P}(Y = 1|X = x) = \frac{\exp\big(\beta_0 + x^T\beta\big)}{1 + \exp\big(\beta_0 + x^T\beta\big)} \equiv \pi_1(x, \beta_0, \beta). \tag{46}$$

In other words, given $X = x$, Y is Bernoulli with mean $\pi_1(x, \beta_0, \beta)$. We can write the model as

$$\text{logit}\big(\mathbb{P}(Y = 1|X = x)\big) = \beta_0 + x^T\beta \tag{47}$$

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. □

The parameters β_0 and $\beta = (\beta_1, \ldots, \beta_d)^T$ 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

$$\begin{aligned} \ell(\beta_0, \beta) &= \sum_{i=1}^{n}\Big\{Y_i \log \pi(x_i, \beta_0, \beta) - (1 - y_i)\log(1 - \pi(x_i, \beta_0, \beta)\Big\} &(48)\\ &= \sum_{i=1}^{n}\Big\{Y_i(\beta_0 + x_i^T\beta) - \log\big(1 + \exp(\beta_0 + x_i^T\beta)\big)\Big\}. &(49)\end{aligned}$$

The maximum conditional likelihood estimators $\widehat{\beta}_0$ and $\widehat{\beta}$ 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:

$$x_i \leftarrow (1, x_i^T)^T \text{ and } \beta \leftarrow (\beta_0, \beta^T)^T. \tag{50}$$

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 kth iterate $\widehat{\beta}^{(k)}$ by

$$\widehat{\beta}^{(k+1)} \leftarrow \widehat{\beta}^{(k)} - \left(\frac{\partial^2 \ell(\widehat{\beta}^{(k)})}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\widehat{\beta}^{(k)})}{\partial \beta}. \tag{51}$$

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

$$\frac{\partial \ell(\widehat{\beta}^{(k)})}{\partial \beta} = \sum_{i=1}^{n} (\pi(x_i, \widehat{\beta}^{(k)}) - Y_i) X_i \text{ and } \frac{\partial^2 \ell(\widehat{\beta}^{(k)})}{\partial \beta \partial \beta^T} = -\mathbb{X}^T \mathbb{W} \mathbb{X} \tag{52}$$

where $\mathbb{W} = \text{diag}(w_{11}^{(k)}, w_{22}^{(k)}, \ldots, w_{dd}^{(k)})$ is a diagonal matrix with

$$w_{ii}^{(k)} = \pi(x_i, \widehat{\beta}^{(k)}) \big(1 - \pi(x_i, \widehat{\beta}^{(k)})\big). \tag{53}$$

Let $\pi_1^{(k)} = \big(\pi_1(x_1, \widehat{\beta}^{(k)}), \ldots, \pi_1(x_n, \widehat{\beta}^{(k)})\big)^T$, (51) can be written as

$$\begin{aligned} \widehat{\beta}^{(k+1)} &= \widehat{\beta}^{(k)} + (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T (y - \pi_1^{(k)}) && (54) \\ &= (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T \mathbb{W} \big(\mathbb{X} \widehat{\beta}^{(k)} + \mathbb{W}^{-1} (y - \pi_1^{(k)}) \big) && (55) \\ &= (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T \mathbb{W} z^{(k)} && (56) \end{aligned}$$

where $z^{(k)} \equiv (z_1^{(k)}, \ldots, z_n^{(k)})^T = \mathbb{X}^T \widehat{\beta}^{(k)} + \mathbb{W}^{-1}(y - \pi_1^{(k)})$ with

$$z_i^{(k)} = \log \left(\frac{\pi_1(x_i, \widehat{\beta}^{(k)})}{1 - \pi_1(x_i, \widehat{\beta}^{(k)})} + \frac{y_i - \pi_1(x_i, \widehat{\beta}^{(k)})}{\pi_1(x_i, \widehat{\beta}^{(k)})(1 - \pi_1(x_i, \widehat{\beta}^{(k)}))} \right). \tag{57}$$

Given the current estimate $\widehat{\beta}^{(k)}$, the above Newton iteration forms a quadratic approximation to the negative log-likelihood using Taylor expansion at $\widehat{\beta}^{(k)}$:

$$-\ell(\beta) = \underbrace{\frac{1}{2}(z - \mathbb{X}\beta)^T \mathbb{W} (z - \mathbb{X}\beta)}_{\ell_Q(\beta)} + \text{constant}. \tag{58}$$

The update equation (56) corresponds to solving a quadratic optimization

$$\widehat{\beta}^{(k+1)} = \underset{\beta}{\text{argmin}}\, \ell_Q(\beta). \tag{59}$$

We then get an iterative algorithm called *iteratively reweighted least squares*. See Figure 7.

Iteratively Reweighted Least Squares Algorithm

Choose starting values $\widehat{\beta}^{(0)} = (\widehat{\beta}_0^{(0)}, \widehat{\beta}_1^{(0)}, \ldots, \widehat{\beta}_d^{(0)})^T$ and compute $\pi_1(x_i, \widehat{\beta}^{(0)})$ using Equation (46), for $i = 1, \ldots, n$ with β_j replaced by its initial value $\widehat{\beta}_j^{(0)}$.

For $k = 1, 2, \ldots$, iterate the following steps until convergence.

1. Calculate $z_i^{(k)}$ according to (57) for $i = 1, \ldots, n$.
2. Calculate $\widehat{\beta}^{(k+1)}$ according to (56). This corresponds to doing a weighted linear regression of z on $\mathbb{X}$.
3. Update the $\pi(x_i, \widehat{\beta})$'s using (46) with the current estimate of $\widehat{\beta}^{(k+1)}$.

Figure 7: Finding the Logistic Regression MLE.

We can get the estimated standard errors of the final solution $\widehat{\beta}$. For the kth iteration, recall that the Fisher information matrix $I(\widehat{\beta}^{(k)})$ takes the form

$$I(\widehat{\beta}^{(k)}) = -\mathbb{E}\left(\frac{\partial^2 \ell(\widehat{\beta}^{(k)})}{\partial\beta\partial\beta^T}\right) \approx \mathbb{X}^T\mathbb{W}\mathbb{X}, \tag{60}$$

we estimate the standard error of $\widehat{\beta}_j$ as the jth diagonal element of $I(\widehat{\beta})^{-1}$.

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	$\widehat{\beta}_j$	*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,

$$\mathbb{P}(Y = 1|X = x) = \frac{p(x|Y = 1)\pi_1}{p(x|Y = 1)\pi_1 + p(x|Y = 0)(1 - \pi_1)} \tag{61}$$

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

$$\begin{aligned} \log\left(\frac{\mathbb{P}(Y = 1|X = x)}{\mathbb{P}(Y = 0|X = x)}\right) &= \log\left(\frac{\pi}{1 - \pi}\right) - \frac{1}{2}(\mu_0 + \mu_1)^T\Sigma^{-1}(\mu_1 - \mu_0) && (62)\\ &\quad + x^T\Sigma^{-1}(\mu_1 - \mu_0) && (63)\\ &\equiv \alpha_0 + \alpha^T x. && (64) \end{aligned}$$

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

$$\prod_{i=1}^{n} p(X_i, Y_i) = \underbrace{\prod_{i=1}^{n} p(X_i|Y_i)}_{\text{Gaussian}} \underbrace{\prod_{i=1}^{n} p(Y_i)}_{\text{Bernoulli}}. \tag{65}$$

In logistic regression we maximize the conditional likelihood $\prod_{i=1}^{n} p(Y_i|X_i)$ but ignore the second term $p(X_i)$:

$$\prod_{i=1}^{n} p(X_i, Y_i) = \underbrace{\prod_{i=1}^{n} p(Y_i|X_i)}_{\text{logistic}} \underbrace{\prod_{i=1}^{n} p(X_i)}_{\text{ignored}}. \tag{66}$$

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

$$\widehat{\beta}_0, \widehat{\beta} = \underset{\beta_0,\beta}{\operatorname{argmin}} \Big\{ -\ell(\beta_0, \beta) + \lambda \|\beta\|_1 \Big\}. \tag{67}$$

Similarly, the *ridge logistic regression* estimator is an ℓ_2-regularized conditional log-likelihood estimator

$$\widehat{\beta}_0, \widehat{\beta} = \underset{\beta_0,\beta}{\operatorname{argmin}} \Big\{ -\ell(\beta_0, \beta) + \lambda \|\beta\|_2^2 \Big\}. \tag{68}$$

The algorithm for logistic ridge regression only requires a simple modification of the iteratively reweighed 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 kth iteration, we first form a quadratic approximation to the negative log-likelihood $\ell(\beta_0, \beta)$ based on the current estimates $\widehat{\beta}^{(k)}$.

$$-\ell(\beta_0, \beta) = \underbrace{\frac{1}{2} \sum_{i=1}^{n} w_{ii} \big(z_i^{(k)} - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \big)^2}_{\ell_Q(\beta_0, \beta)} + \text{constant}. \tag{69}$$

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

$$\widehat{\beta}^{(k+1)}, \widehat{\beta}^{(k+1)} = \underset{\beta_0,\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} w_{ii} \big(z_i^{(k)} - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \big)^2 + \lambda \|\beta\|_1 \right\}. \tag{70}$$

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 $\widehat{\beta}^{(0)} = (\widehat{\beta}_0^{(0)}, \widehat{\beta}_1^{(0)}, \ldots, \widehat{\beta}_d^{(0)})^T$

(Outer loop) For $k = 1, 2, \ldots$, iterate the following steps until convergence.

1. For $i = 1, \ldots, n$, 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 $\ell = 1, \ldots d$.

3. (Inner loop) iterate the following steps until convergence

   For $j \in \{1, \ldots, d\}$

   (a) For $i = 1, \ldots, n$, 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) $\alpha_j = \text{sign}\big(u_j^{(k)}\big) \left[\frac{|u_j^{(k)}| - \lambda}{v_j^{(k)}} \right]_+$.

4. $\widehat{\beta}_0^{(k+1)} = \alpha_0$ and $\widehat{\beta}_\ell^{(k+1)} = \alpha_\ell$ for $\ell = 1, \ldots d$.

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

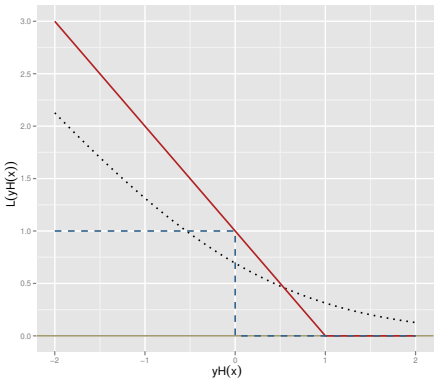


Figure 9: The 0-1 classification loss (blue dashed line), hinge loss (red solid line) and logistic loss (black dotted line).

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* $L_{\text{hinge}}\big(y_i, H(x_i)\big) \equiv \big[1 - Y_i H(X_i)\big]_+$ 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 $\widehat{h}(x) = I\big(\widehat{H}(x) > 0\big)$ where the hyperplane $\widehat{H}(x) = \widehat{\beta}_0 + \widehat{\beta}^T x$ is obtained by minimizing

$$\sum_{i=1}^{n} \big[1 - Y_i H(X_i)\big]_+ + \frac{\lambda}{2}\|\beta\|_2^2 \tag{71}$$

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 $\mathbb{E}\big(1 - YH(X)\big)_+$ 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

$$-\sum_{i \in \mathcal{M}} Y_i H(X_i).$$

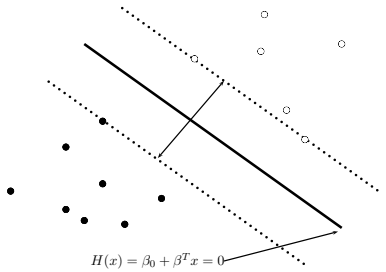


Figure 10: The hyperplane $H(x)$ has the largest margin of all hyperplanes that separate the two classes.

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:

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \longleftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} Y_i X_i \\ Y_i \end{pmatrix}$$

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:

$$\min_{\beta_0,\beta} \quad \left\{ \frac{1}{2}\|\beta\|_2^2 + \frac{1}{\lambda}\sum_{i=1}^{n} \xi_i \right\} \tag{72}$$

$$\text{subject to} \quad \forall i, \xi_i \geq 0 \text{ and } \xi_i \geq 1 - y_i H(x_i). \tag{73}$$

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*

$$\widehat{\alpha} = \underset{\alpha\in\mathbb{R}^n}{\operatorname{argmax}}\Big\{\sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{k=1}^n \alpha_i\alpha_k Y_i y_k \langle X_i, x_k\rangle\Big\} \tag{74}$$

$$\text{subject to } 0 \le \alpha_1, \ldots, \alpha_n \le \frac{1}{\lambda} \text{ and } \sum_{i=1}^n \alpha_i y_i = 0, \tag{75}$$

with the primal-dual relationship $\widehat{\beta} = \sum_{i=1}^n \widehat{\alpha} y_i x_i$. *We also have*

$$\widehat{\alpha}_i\Big(1 - \xi_i - y_i\big(\widehat{\beta}_0 + \widehat{\beta} x_i\big)\Big) = 0, \quad i = 1, \ldots, n. \tag{76}$$

Proof. Let $\alpha_i, \gamma_i \ge 0$ be the Lagrange multipliers. The Lagrangian function can be written as

$$L(\xi, \beta, \beta_0, \alpha, \gamma) = \frac{1}{2}\|\beta\|_2^2 + \frac{1}{\lambda}\sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i\big(1 - \xi_i - y_i H(x_i)\big) - \sum_{i=1}^n \gamma_i\xi_i. \tag{77}$$

The Karush-Kuhn-Tucker conditions are

$$\forall i,\ \alpha_i \ge 0, \gamma_i \ge 0,\ \xi_i \ge 0 \text{ and } \xi_i \ge 1 - y_i H(x_i), \tag{78}$$

$$\beta = \sum_{i=1}^n \alpha_i y_i x_i, \quad \sum_{i=1}^n \alpha_i y_i = 0 \ \text{ and } \ \alpha_i + \gamma_i = 1/\lambda, \tag{79}$$

$$\forall i,\ \alpha_i\big(1 - \xi_i - y_i H(x_i)\big) = 0 \ \text{ and } \ \gamma_i\xi_i = 0. \tag{80}$$

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). □

The dual problem (74) is easier to solve than the primal problem (71). The data points (X_i, y_i) for which $\widehat{\alpha}_i > 0$ are called *support vectors*. By (76) and (72), for all the data points (x_i, y_i) satisfying $y_i\big(\widehat{\beta}_0 + \widehat{\beta}^T x_i\big) > 1$, there must be $\widehat{\alpha}_i = 0$. The solution for the dual problem is sparse. From the first equality in (79), we see that the final estimate $\widehat{\beta}$ 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

$$\widehat{\beta}_0 = -Y_i X_i^T \widehat{\beta}. \tag{81}$$

Therefore, once $\widehat{\beta}$ is given, we could calculate $\widehat{\beta}_0$ using any margin point (X_i, y_i).

Example 12 *We consider classifying two types of irises, versicolor and viriginica. 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.

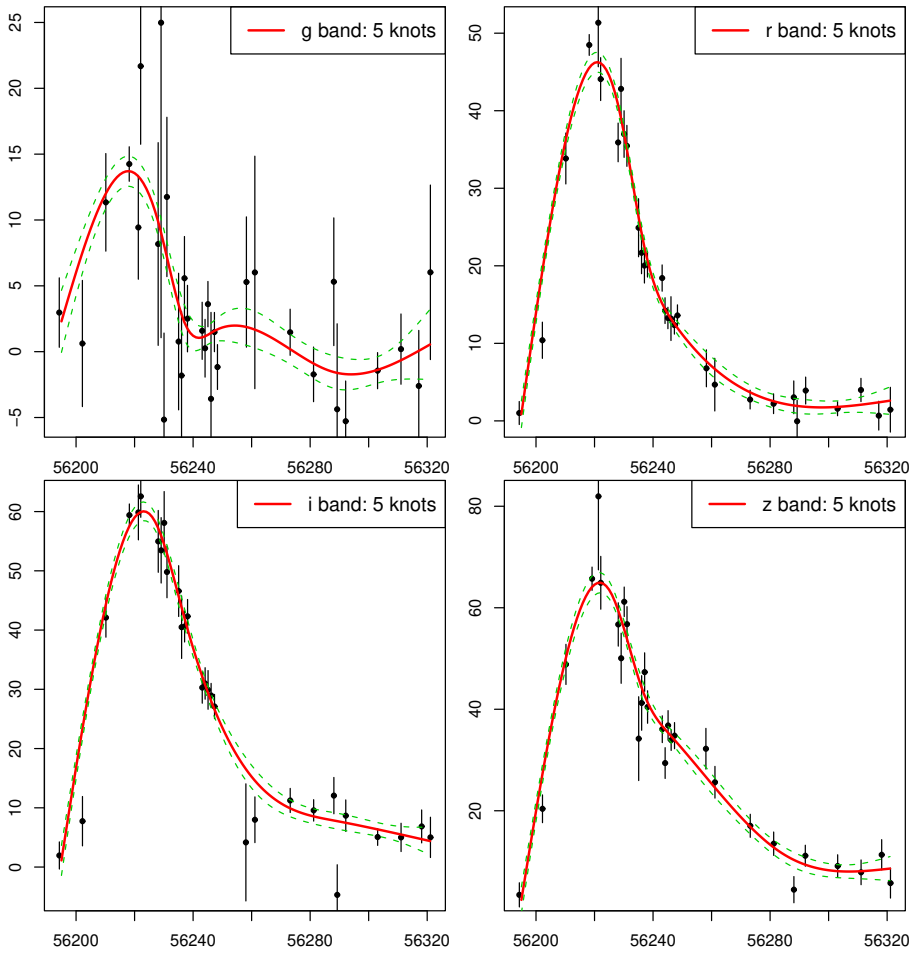
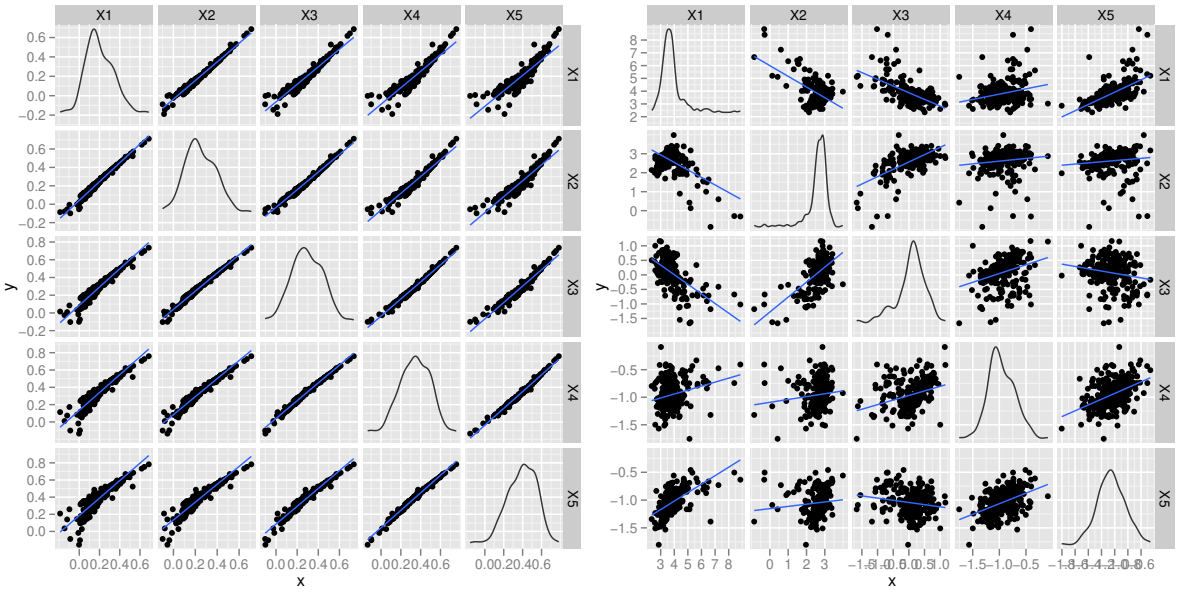


Figure 12: Four filters (g, r, i, z-bands) corresponding to a type Ia supernova *DES-SN000051*. For each band, a weighted regression spline fit (solid red) with the corresponding standard error curves (dashed green) is provided. The black points with bars represent the flux values and their estimated 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.



time domain frequency domain

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 $X_1, \ldots, X_d$ ($d = 100$), Their corresponding frequency domain features $\widetilde{X}_1, \ldots, \widetilde{X}_d$ can be

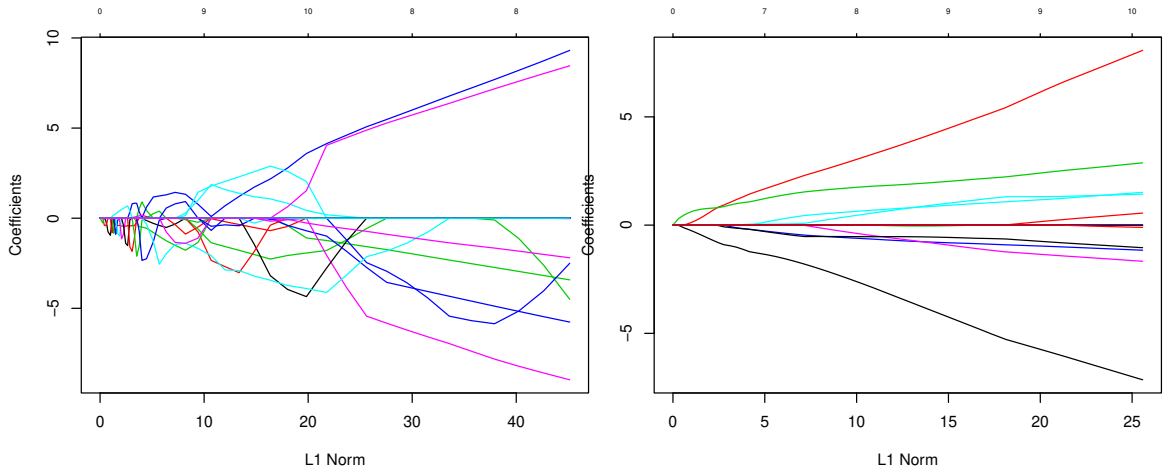
written as

$$\widetilde{X}_j = \frac{2}{d}\sum_{k=1}^{d} X_k \cos\Big[\frac{\pi}{d}\big(k-\frac{1}{2}\big)(j-1)\Big] \text{ for } j = 1,\ldots,d. \tag{82}$$

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.

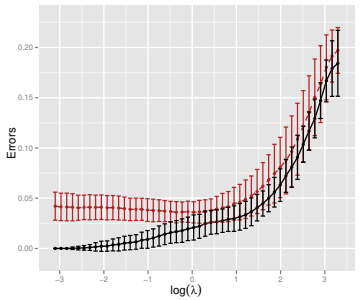
Figure 14 illustrates the regularization paths of sparse logistic regression using the ime-domain and frequency-domain features. A regularization path provides the coefficient value of each feature over all regularization parameters. Since the time-domain features are highly correlated, the corresponding regularization path is quite irregular. In contrast, the paths for the frequency-domain features behave stably.



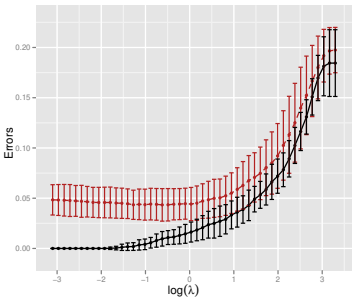
time domain frequency domain

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.

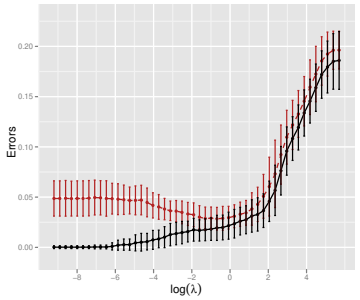
Figure 15 compares the classification performance of all these methods. The results show that classification in the frequency domain is not helpful. The regularization paths of the SVM are the same in both the time and frequency domains. This is expected since the discrete cosine transformation is an orthonormal transformation, which corresponds to rotatating the data in the feature space while preserving their Euclidean distances and inner products. It is easy



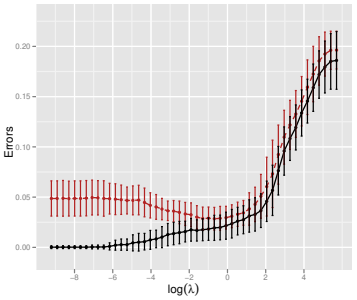
Sparse LR (time domain)



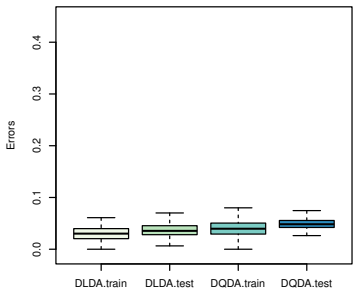
Sparse LR (frequency domain)



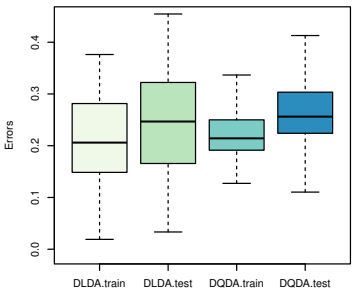
SVM (time domain)



SVM (frequency domain)



Discriminant Analysis (time domain)



Discriminant Analysis (frequency domain)

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.

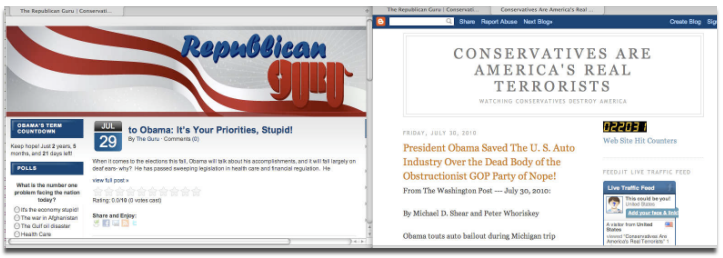
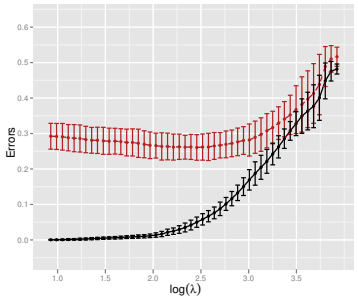


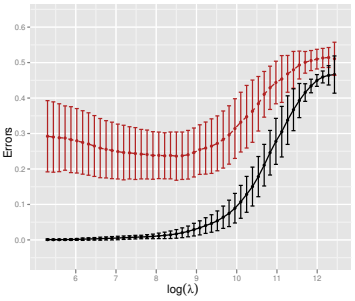
Figure 16: Examples of two political blogs with different orientations, one conservative and the other liberal.

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 $i = 1, \ldots, 403$, its linkage information is represented as a 292-dimensional binary vector $(x_{i1}, \ldots, x_{i292})^T$ where $x_{ij} = 1$ if the ith blog has a link to the jth 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.

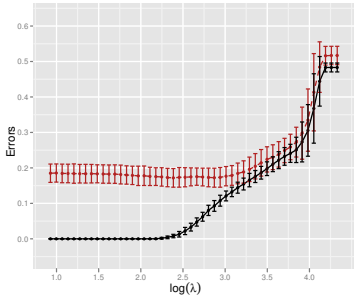
We run the full regularization paths of sparse logistic regression and support vector machines,



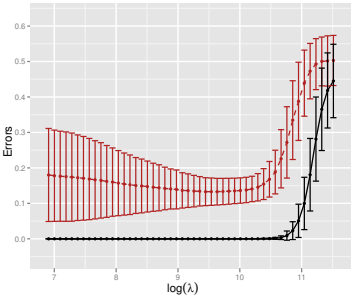
Sparse LR



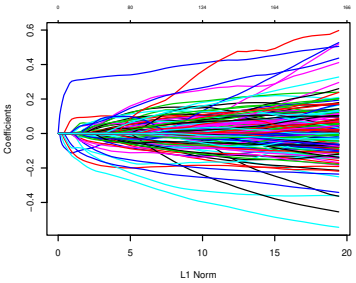
SVM



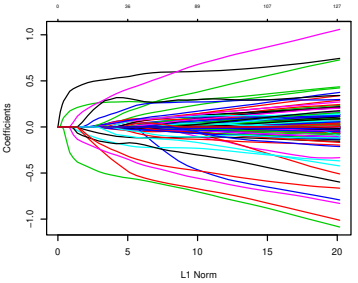
Sparse LR (with linkage information)



SVM (with linkage information)



Sparse LR path (no linkage information)



Sparse LR path (with linkage information)

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 thev 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 $h : \mathcal{X} \to \{0, 1\}$ 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

$$R(h) = \mathbb{P}(Y \neq h(X)) \tag{1}$$

and the *empirical error rate* or *training error rate* based on training data $(X_1, Y_1), \ldots, (X_n, Y_n)$ is

$$\widehat{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} I(h(X_i) \neq Y_i). \tag{2}$$

$R(h)$ is minimized by the *Bayes' rule*

$$h^*(x) = \begin{cases} 1 & \text{if } m(x) > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} = \begin{cases} 1 & \text{if } \frac{p_1(x)}{p_0(x)} > \frac{(1-\pi)}{\pi} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

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, $Y \in \{1, \ldots, k\}$, the Bayes' rule is

$$h^*(x) = \text{argmax}_{1 \leq j \leq k} \pi_j \; p_j(x) = \text{argmax}_{1 \leq j \leq k} m_j(x)$$

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

$$\widehat{h}(x) = \begin{cases} 1 & \text{if } \widehat{m}(x) > \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

For example, we could use the kernel regresson estimator

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

Howeve, 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,*

$$R(\widehat{h}) - R(h^*) \leq 2 \int |\widehat{m}(x) - m(x)| dP(x) \leq 2 \sqrt{\int |\widehat{m}(x) - m(x)|^2 dP(x)}. \tag{5}$$

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 $Y \in \{1, \ldots, k\}$ the plugin rule has the form

$$\widehat{h}(x) = \text{argmax}_j \widehat{m}_j(x)$$

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

$$\widehat{h}(x) = \begin{cases} 1 & \text{if } \frac{\widehat{p}_1(x)}{\widehat{p}_0(x)} > \frac{(1-\widehat{\pi})}{\widehat{\pi}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

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 $X_i = (X_{i1}, \ldots, X_{id})^T$ 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 $Y \in \{1, \ldots, k\}$ then we estimate the k densities $\widehat{p}_j(x) = p(x|Y = j)$ and the classifier is

$$\widehat{h}(x) = \text{argmax}_j \ \widehat{\pi}_j \ \widehat{p}_j(x)$$

where $\widehat{\pi}_j = n^{-1} \sum_{i=1}^n I(Y_i = j)$.

4 Nearest Neighbors

The *k-nearest neighbor classifier* is

$$h(x) = \begin{cases} 1 & \sum_{i=1}^n w_i(x) I(Y_i = 1) > \sum_{i=1}^n w_i(x) I(Y_i = 0) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

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 estsimator

$$\widehat{m}(x) = \frac{\sum_{i=1}^n Y_i \ I(||X_i \leq x|| \leq d_k(x))}{\sum_{i=1}^n I(||X_i \leq x|| \leq d_k(x))}$$

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 $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$. Then we have that

$$\lim_{n\to\infty} \mathbb{E}(L_n) = \mathbb{E}(2m(X)(1 - m(X))) \equiv R_{(1)}. \tag{8}$$

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,

$$\lim_{n\to\infty} \mathbb{E}(L_n) = R_{(k)} \tag{9}$$

where

$$R_{(k)} = \mathbb{E}\left(\sum_{j=0}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) m^j(X)(1 - m(X))^{k-j} \left[m(X)I(j < k/2) + (1 - m(X))I(j > k/2)\right]\right).$$

Theorem 2 (Devroye et al 1996) *For all odd k,*

$$R_* \leq R_{(k)} \leq R_* + \frac{1}{\sqrt{ke}}. \tag{10}$$

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 $k \to \infty$ and $k/n \to 0$. For every $\epsilon > 0$ the following is true. For all large n,*

$$\mathbb{P}(R(\widehat{h}) - R_* > \epsilon) \leq e^{-n\epsilon^2/(72\gamma_d^2)}$$

where $\widehat{h}_n$ 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*

$$P\left(\{x:\ |m(x)-(1/2)|\leq t\}\right)\leq Ct^{\beta}$$

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 $B=\{u:\ ||x-u||\leq r\}$, $B^0=\{u:\ ||x-u||<r\}$ 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 $1-\delta$,

$$P(\widehat{h}(X)\leq h_*(X))\leq \delta C\left(\frac{\log(1/\delta)}{n}\right)^{\frac{\alpha\beta}{2\alpha+1}}.$$

If $k\asymp n^{\frac{2\alpha}{2\alpha+1}}$ then

$$R(\widehat{h})-R(h_*)\preceq n^{-\frac{\alpha(\beta+1)}{2\alpha+1}}.$$

4.1 Partitions and Trees

As with nonparametric regression, simple and interpretable classifiers can be derived by partitioning the range of X. Let $\Pi_n=\{A_1,\ldots,A_N\}$ be a partition of $\mathcal{X}$. Let A_j be the partition element that contains x. Then $\widehat{h}(x)=1$ if $\sum_{X_i\in A_j}Y_i\geq\sum_{X_i\in A_j}(1-Y_i)$ and $\widehat{h}(x)=0$ otherwise. This is nothing other than the plugin classifier based on the partition regression estimator

$$\widehat{m}(x)=\sum_{j=1}^{N}\overline{Y}_j\,I(x\in A_j)$$

where $\overline{Y}_j=n_j^{-1}\sum_{i=1}^{n}Y_iI(X_i\in A_j)$ is the average of the Y_i's in A_j and $n_j=\#\{X_i\in A_j\}$. (We define $\overline{Y}_j$ to be 0 if $n_j=0$.)

Recall from the results on regression that if

$$m\in\mathcal{M}=\left\{m:\ |m(x)-m(x)|\leq L||x-z||,\quad x,z,\in\mathbb{R}^d\right\}\tag{11}$$

and the binwidth b satsifies $b\asymp n^{-1/(d+2)}$ then

$$\mathbb{E}||\widehat{m}-m||_P^2\leq\frac{c}{n^{2/(d+2)}}.\tag{12}$$

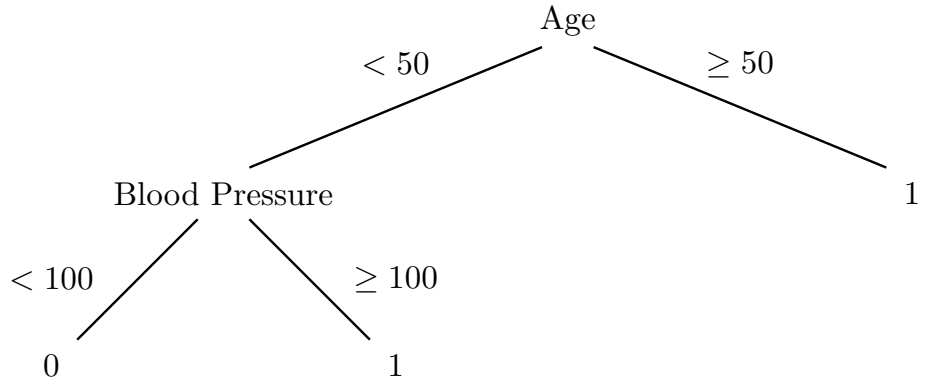


Figure 1: A simple classification tree.

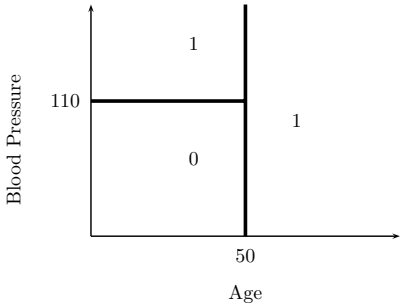


Figure 2: Partition representation of classification tree.

From (5), we conclude that $R(\widehat{h}) - R(h_*) = O(n^{-1/(d+2)})$. 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, $X_1 =$ age and $X_2 =$ blood pressure. Figure 1 shows a classification tree using these variables.

The tree is used in the following way. If a subject has Age ≥ 50 then we classify him as $Y = 1$. If a subject has 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 $Y_i = j$:

$$r_s(j) = \frac{\sum_{i=1}^n I(Y_i = j, X_i \in A_s)}{\sum_{i=1}^n I(X_i \in A_s)} \tag{13}$$

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

$$\gamma_s = 1 - \sum_{j=0}^{1} r_s(j)^2. \tag{14}$$

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 $Y \in \{1, \ldots, K\}$. We define the impurity by

$$\gamma_s = 1 - \sum_{j=1}^{k} r_s^2(j) \tag{15}$$

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

$$R_n(\mathcal{P}) = \inf_{\widehat{h}} \sup_{P \in \mathcal{P}} \left(R(\widehat{h}) - R_n^* \right) \tag{16}$$

where $R(\widehat{h}) = \mathbb{P}(Y \neq \widehat{h}(X))$, R_n^* is the Bayes error and the infimum is over all classifiers constructed from the data $(X_1, Y_1), \ldots, (X_n, Y_n)$. Recall that

$$R(\widehat{h}) - R(h^*) \leq 2 \sqrt{\int |\widehat{m}(x) - m(x)|^2 dP(x)}$$

Class	Rate	Condition
$\mathcal{E}(\alpha)$	$n^{-\alpha/(2\alpha+d)}$	$\alpha > 1/2$
BV	$n^{-1/3}$	
MI	$\sqrt{\log n/n}$	
$L(\alpha, q)$	$n^{-\alpha/(2\alpha+1)}$	$\alpha > (1/q - 1/2)_+$
$B^{\alpha}_{\sigma,q}$	$n^{-\alpha/(2\alpha+d)}$	$\alpha/d > 1/q - 1/2$
Neural nets	see text	

Table 1: Minimax Rates of Convergence for Classification.

Thus $R_n(\mathcal{P}) \leq 2\sqrt{\widetilde{R}_n(\mathcal{P})}$ where $\widetilde{R}_n(\mathcal{P})$ 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 $2\sqrt{\widetilde{R}_n(\mathcal{P})}$? 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 th 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 $B^{\alpha}_{\sigma,q}$ 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 $d \to \infty$, 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 $\widehat{h}(x) = \text{sign}(\widehat{H}(x))$ where $\widehat{H}(x) = \widehat{\beta}_0 + \widehat{\beta}^T x$ and $\widehat{\beta}$ 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 controversey 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. Futher 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: $(X_1, Y_1), \ldots, (X_n, Y_n)$ where $Y_i \in \{-1, +1\}$.
2. Set $w_i = 1/n$ for $i = 1, \ldots, n$.
3. Repeat for $m = 1, \ldots, M$.
   (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 $(X_1, Y_1), \ldots, (X_{2n}, Y_{2n})$. Now randomly split the data into two halves that we denote by

$$\mathcal{D} = \Big\{(\widetilde{X}_1, \widetilde{Y}_1), \ldots, (\widetilde{X}_n, \widetilde{Y}_n)\Big\}, \quad \text{and} \quad \mathcal{E} = \Big\{(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)\Big\}.$$

Construct classifiers $\mathcal{H} = \{h_1, \ldots, h_N\}$ 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 $\widehat{h} = \text{argmin}_{h\in\mathcal{H}}\widehat{R}(h)$.

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; typicaly $K = 10$. 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

$$\mathbb{P}(Y = 1 \,|\, X) \equiv p(X; f) = \frac{\exp\left(\sum_{j=1}^p f_j(X_j)\right)}{1 + \exp\left(\sum_{j=1}^p f_j(X_j)\right)} \tag{17}$$

where $Y \in \{0, 1\}$, and the population log-likelihood is

$$\ell(f) = \mathbb{E}\left[Y f(X) - \log\left(1 + \exp f(X)\right)\right] \tag{18}$$

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.

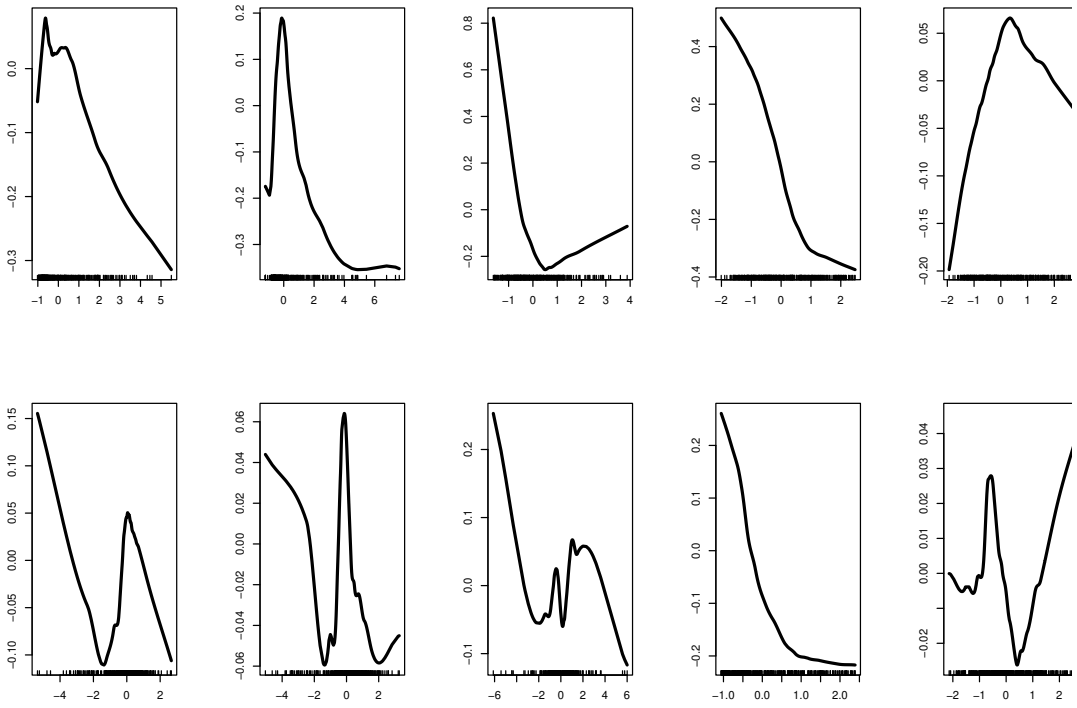


Figure 4: Estimated functions for additive model.

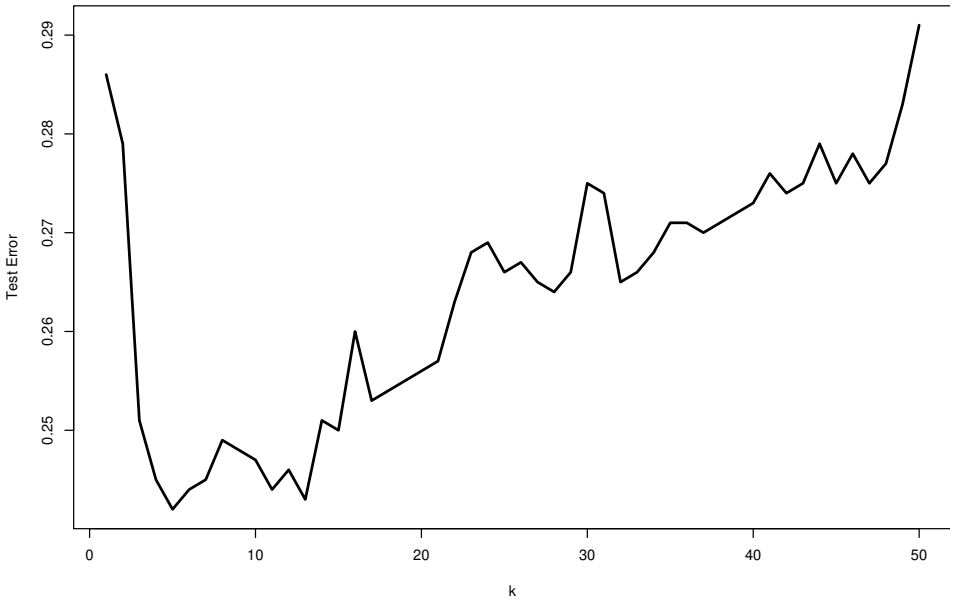


Figure 5: Test error versus k for nearest neighbor estimator.

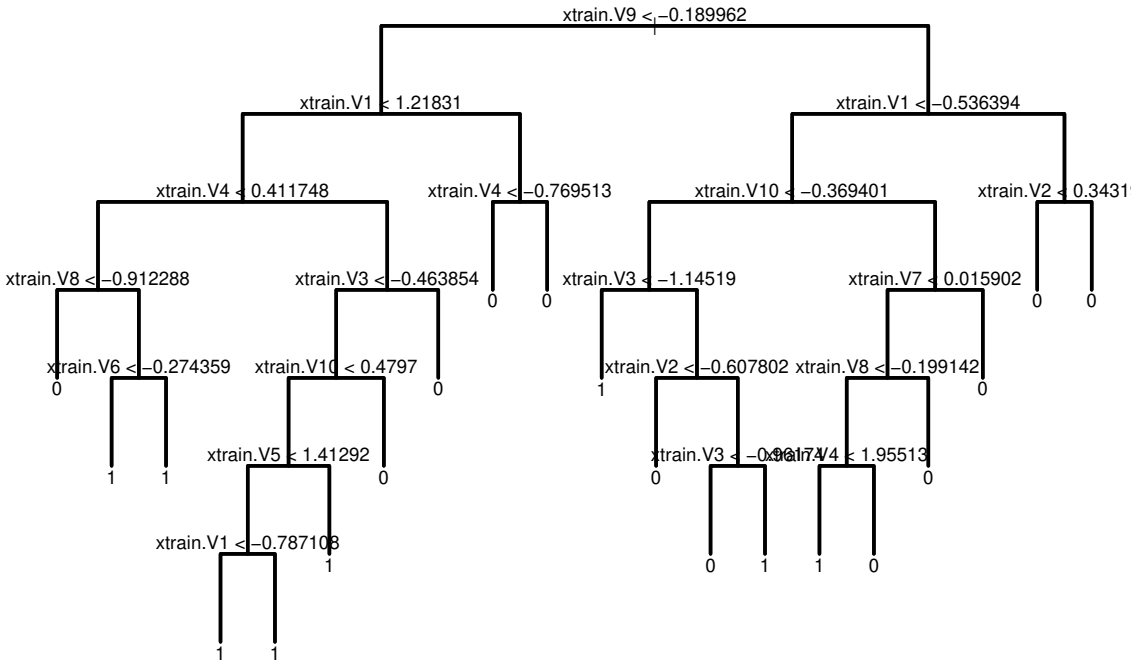


Figure 6: Full tree.

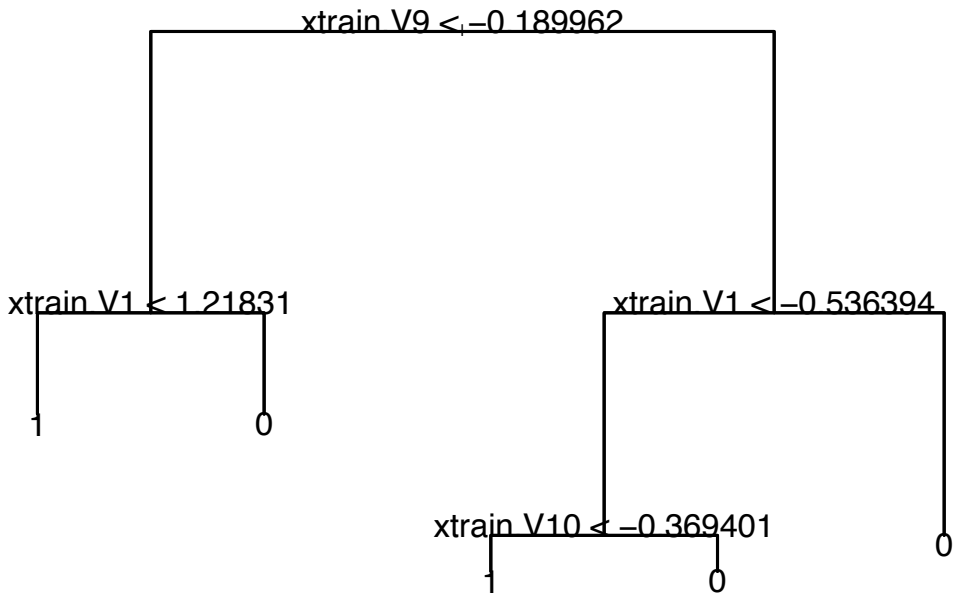


Figure 7: Classification tree. The size of the tree was chcosen by cross-validation.

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 $\widehat{f}$

$$Z_i = \widehat{f}(X_i) + \frac{Y_i - p(X_i; \widehat{f})}{p(X_i; \widehat{f})(1 - p(X_i; \widehat{f}))} \tag{19}$$

and weights $w(X_i) = p(X_i; \widehat{f})(1 - p(X_i; \widehat{f})$, and carries out a weighted backfitting of (Z, X) with weights w. The weighted smooth is given by

$$\widehat{P}_j = \frac{\mathcal{S}_j(wR_j)}{\mathcal{S}_j w}. \tag{20}$$

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 penality can be incorporated, just as for sparse additive models (SpAM) for regression. The Lagrangian is given by

$$\mathcal{L}(f, \lambda) = \mathbb{E}\left[\log\left(1 + e^{f(X)}\right) - Y f(X)\right] + \lambda\left(\sum_{j=1}^{p} \sqrt{\mathbb{E}(f_j^2(X_j))} - L\right) \tag{21}$$

and the stationary condition for component function f_j is $\mathbb{E}\left(p - Y \mid X_j\right) + \lambda v_j = 0$ 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 $\widehat{f}$. This yields the linearized condition $\mathbb{E}\left[w(X)(f(X) - Z) \,|\, X_j\right] + \lambda v_j = 0$. To see this, note that

$$\begin{aligned} 0 &= \mathbb{E}\left(p(X;\widehat{f}) - Y + p(X;\widehat{f})(1 - p(X;\widehat{f}))(f(X) - \widehat{f}(X)) \,|\, X_j\right) + \lambda v_j & (22) \\ &= \mathbb{E}\left[w(X)(f(X) - Z) \,|\, X_j\right] + \lambda v_j & (23) \end{aligned}$$

When $\mathbb{E}(f_j^2) \neq 0$, this implies the condition

$$\left(\mathbb{E}\left(w \,|\, X_j\right) + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}}\right) f_j(X_j) = \mathbb{E}(wR_j \,|\, X_j). \tag{24}$$

In the finite sample case, in terms of the smoothing matrix $\mathcal{S}_j$, this becomes

$$f_j = \frac{\mathcal{S}_j(wR_j)}{\mathcal{S}_j w + \lambda \Big/ \sqrt{\mathbb{E}(f_j^2)}}. \tag{25}$$

If $\|\mathcal{S}_j(wR_j)\| < \lambda$, then $f_j = 0$. Otherwise, this implicit, nonlinear equation for f_j cannot be solved explicitly, so one simply iterates until convergence:

$$f_j \;\leftarrow\; \frac{\mathcal{S}_j(wR_j)}{\mathcal{S}_j w + \lambda\sqrt{n}\,/\|f_j\|}. \tag{26}$$

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 $h_1, \ldots, h_B$. 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 ($\surd$)	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

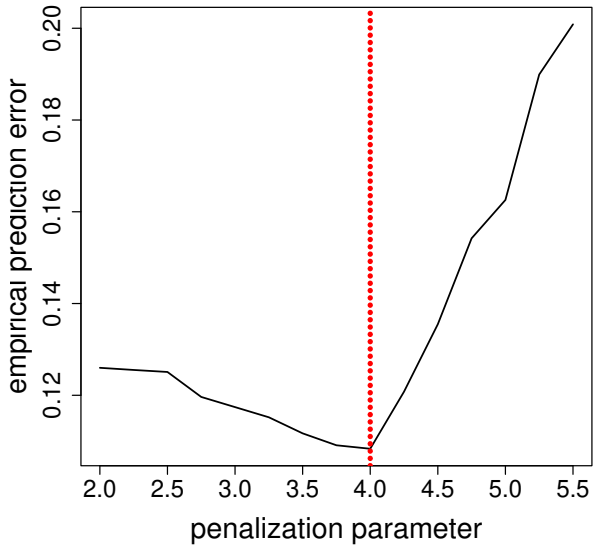


Figure 8: (Email spam) Classification accuracies and variable selection for logistic SpAM.

This is called *bagging* which stands for *bootstrap aggregration*. The basline 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

$$p_f(Y = \ell \,|\, X) = \frac{e^{f_\ell(X)}}{\sum_{m=1}^K e^{f_m(X)}} \quad \ell = 1, \ldots, K \tag{27}$$

where each function has an additive form

$$f_\ell(X) = f_{\ell 1}(X_1) + f_{\ell 2}(X_2) + \cdots + f_{\ell p}(X_p). \tag{28}$$

In Newton's algorithm, we minimize the quadratic approximation to the log-likelihood

$$L(f) \;\approx\; L(\widehat{f}) + \mathbb{E}\left[(Y - \widehat{p})^T (f - \widehat{f})\right] + \frac{1}{2}\mathbb{E}\left[(f - \widehat{f})^T H(\widehat{f})(f - \widehat{f})\right] \tag{29}$$

where $\widehat{p}(X) = (p_{\widehat{f}}(Y = 1 \,|\, X), \ldots, p_{\widehat{f}}(Y = K \,|\, X))$, and $H(\widehat{f}(X))$ is the Hessian

$$H(\widehat{f}) = -\mathrm{diag}(\widehat{p}(X)) + \widehat{p}(X)\widehat{p}(X)^T. \tag{30}$$

Maximizing the right hand size of (29) is equivalent to minimizing

$$-\mathbb{E}\left((Y - \widehat{p})^T (f - \widehat{f})\right) - \mathbb{E}\left(\widehat{f}^T J f\right) + \frac{1}{2}\mathbb{E}\left(f^T J f\right) \tag{31}$$

which is, in turn, equivalent to minimizing the surrogate loss function

$$Q(f, \widehat{f}) \equiv \;=\; \frac{1}{2}\mathbb{E}\left(\|Z - Af\|_2^2\right). \tag{32}$$

where $J = -H(\widehat{f})$, $A = J^{1/2}$, and Z is defined by

$$Z \;=\; J^{-1/2}(Y - \widehat{p}) + J^{1/2}\widehat{f} \tag{33}$$

$$\;=\; A^{-1}(Y - \widehat{p}) + A\widehat{f}. \tag{34}$$

The above calculation can be reexpressed as follows, which leads a multiclass backfitting algorithm. The difference in log-likelihoods for functions $\{\widehat{f}_\ell\}$ and $\{f_\ell\}$ is, to second order,

$$\mathbb{E}\left[\sum_{\ell=0}^{K-1} p_\ell(X)\left(\widehat{f}_\ell(X) - \sum_{k=0}^{K-1} p_k(X)\widehat{f}_k(X) + \frac{Y_\ell - p_\ell(X)}{p_\ell(X)} - f_\ell(X) + \sum_{k=0}^{K-1} p_k(X) f_k(X)\right)^2\right] \tag{35}$$

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 $f_k = \widehat{f}_k$ for $k \neq \ell$, this simplifies to

$$\mathbb{E}\left[p_\ell(1-p_\ell)^2\left(\widehat{f}_\ell + \frac{Y_\ell - p_\ell}{p_\ell(1-p_\ell)} - f_\ell\right)^2 + \sum_{k\neq\ell} p_k p_\ell^2\left(\widehat{f}_\ell + \frac{p_k - Y_k}{p_k p_\ell} - f_\ell\right)^2\right]. \tag{36}$$

After some algebra, this can be seen to be the same as the usual objective function in the binary case, where we take $\widehat{f}_0 = 1$ and $\widehat{f}_1$ arbitrary.

Now assume f_ℓ (and $\widehat{f}_\ell$) 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:

$$f_{\ell j}(X_j) \leftarrow \frac{\mathbb{E}\left[p_\ell(1-p_\ell)\left(\widehat{f}_\ell - \sum_{k\neq j} f_{\ell k} + \frac{Y_\ell - p_\ell}{p_\ell(1-p_\ell)}\right) \,|\, X_j\right]}{\mathbb{E}\left[p_\ell(1-p_\ell) \,|\, X_j\right]}. \tag{37}$$

We approximate the conditional expectations by smoothing, as usual:

$$f_{\ell j}(x_j) \leftarrow \frac{\mathcal{S}_j(x_j)^T\left(w_\ell(X)\, R_{\ell j}(X)\right)}{\mathcal{S}_j(x_j)^T\left(w_\ell(X)\right)} \tag{38}$$

where

$$R_{\ell j}(X) = \widehat{f}_\ell(X) - \sum_{k\neq j} f_{\ell k}(X_k) + \frac{Y_\ell - p_\ell(X)}{p_\ell(X)(1-p_\ell(X))} \tag{39}$$

$$w_\ell(X) = p_\ell(X)(1-p_\ell(X)). \tag{40}$$

This is the same as in binary logistic regression. We thus have the following algorithm:

Multiclass Logistic Backfitting

1. Initialize $\{\widehat{f}_\ell = 0\}$, and set $Z(X) = K$.
2. Iterate until convergence:

   For each $\ell = 0, 1, \ldots, K-1$

   A. Initialize $f_\ell = \widehat{f}_\ell$

   B. Iterate until convergence:

For each $j = 1, 2, \ldots, p$

$$
\begin{aligned}
f_{\ell j}(x_j) &\leftarrow \frac{\mathcal{S}_j(x_j)^T \left(w_\ell(X)\, R_{\ell j}(X)\right)}{\mathcal{S}_j(x_j)^T \left(w_\ell(X)\right)} \text{ where} \\
R_{\ell j}(X) &= \widehat{f}_\ell(X) - \sum_{k \neq j} f_{\ell k}(X_k) + \frac{Y_\ell - p_\ell(X)}{p_\ell(X)(1 - p_\ell(X))} \\
w_\ell(X) &= p_\ell(X)(1 - p_\ell(X)).
\end{aligned}
$$

C. Update $Z(X) \leftarrow Z(X) - e^{\widehat{f}_\ell(X)} + e^{f_\ell(X)}$.

D. Set $\widehat{f}_\ell \leftarrow f_\ell$.

Incrementally updating the normalizing constants (step C) is important so that the probabilties $p_\ell(X) = e^{\widehat{f}_\ell(X)}/Z(X)$ 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 $\Pi_n = \{A_1, \ldots, A_N\}$ be a partition of $\mathcal{X}$. Let A_j be the partition element that contains x. Then $\widehat{h}(x) = 1$ if $\sum_{X_i \in A_j} Y_i \geq \sum_{X_i \in A_j} (1 - Y_i)$ and $\widehat{h}(x) = 0$ otherwise. This is nothing other than the plugin classifier based on the partition regression estimator

$$\widehat{m}(x) = \sum_{j=1}^{N} \overline{Y}_j \, I(x \in A_j)$$

where $\overline{Y}_j = n_j^{-1} \sum_{i=1}^{n} Y_i I(X_i \in A_j)$ is the average of the Y_i's in A_j and $n_j = \#\{X_i \in A_j\}$. (We define $\overline{Y}_j$ 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

$$\mathbb{E}||\widehat{m} - m||_P^2 \leq \frac{c}{n^{2/(d+2)}}. \tag{1}$$

We conclude that the corresponding classification risk satisfies $R(\widehat{h}) - R(h_*) = O(n^{-1/(d+2)})$.

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 Age ≥ 50 then we classify him as $Y = 1$. If a subject has 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 $\overline{Y}_1$ be the mean of the Y_i's in A_1 and let $\overline{Y}_2$ be the mean of the Y_i's in A_2.

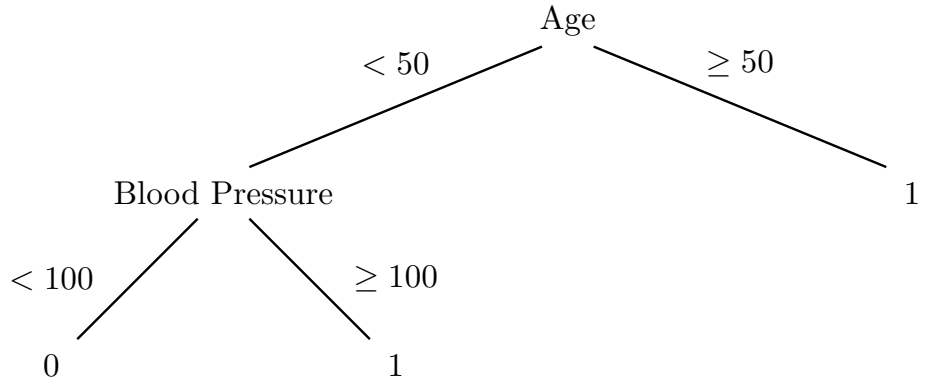


Figure 1: A simple classification tree.

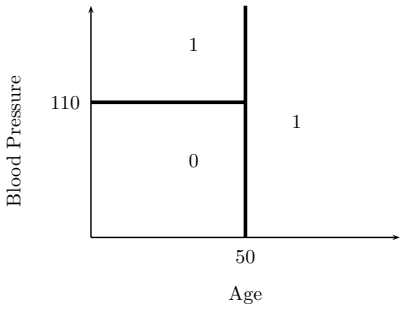


Figure 2: Partition representation of classification tree.

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

$$\gamma_s = 1 - [\overline{Y}_s^2 + (1 - \overline{Y}_s)^2]. \tag{2}$$

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 $\widehat{m}(x)$ which is the mean of Y_i's in that leaf. For classification, we take $\widehat{h}(x) = I(\widehat{m}(x) > 1/2)$. 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 $h_1, \ldots, h_B$. (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.

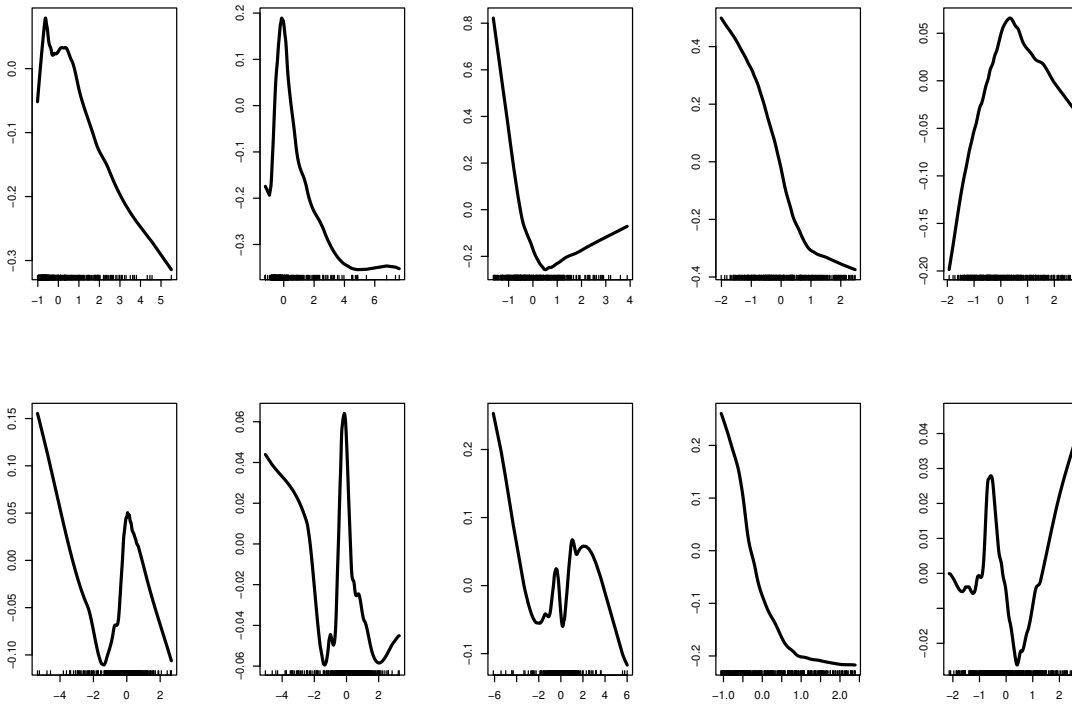


Figure 3: Estimated functions for additive model.

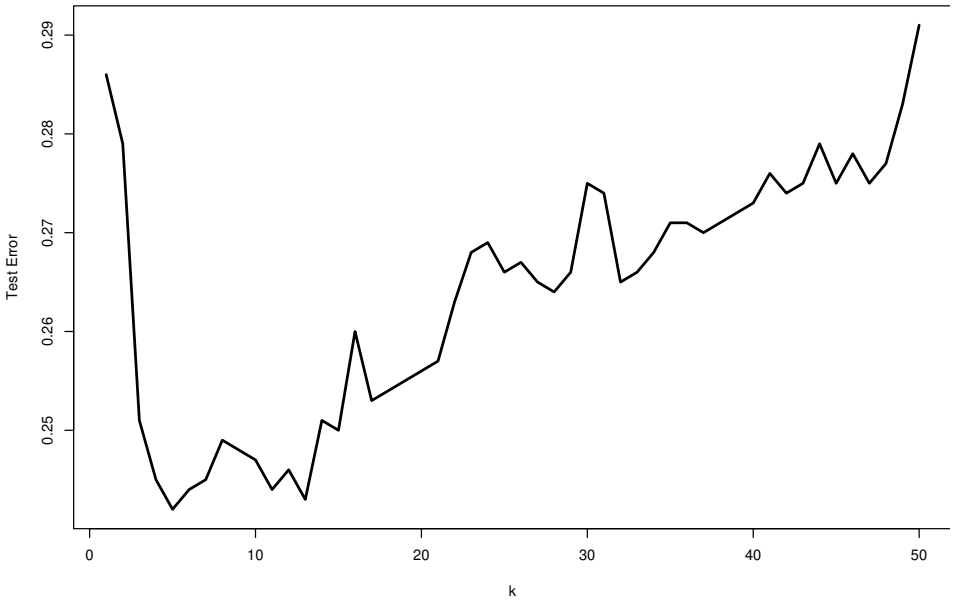


Figure 4: Test error versus k for nearest neighbor estimator.

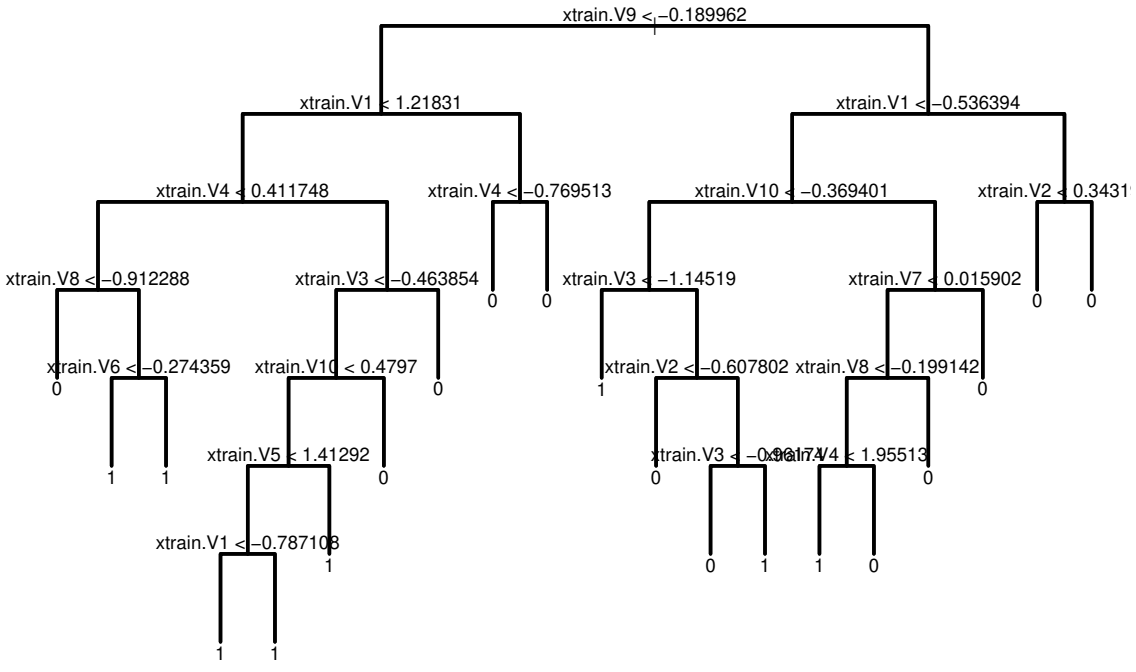


Figure 5: Full tree.

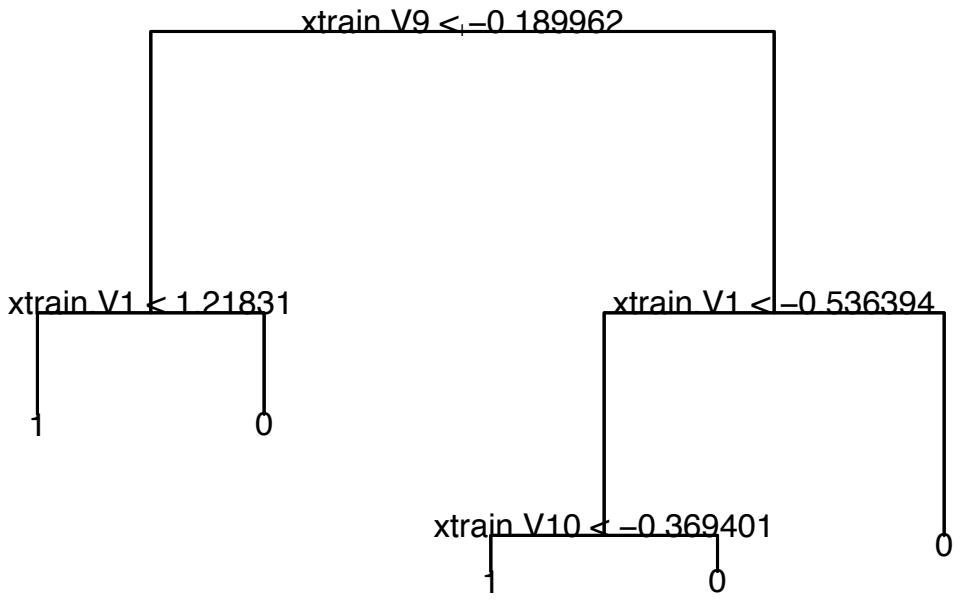


Figure 6: Classification tree. The size of the tree was chosen by cross-validation.

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 $\widehat{\theta}_n = I(\overline{Y}_n \leq x)$. 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 $\widehat{\theta}_n$ converges to $I(Z \leq c)$ where $Z \sim N(0,1)$. So the limiting mean and variance of $\widehat{\theta}_n$ are $\Phi(c)$ and $\Phi(c)(1-\Phi(c))$. Now the bootstrap distribution of $\overline{Y}^*$ (conditional on $Y_1, \ldots, Y_n$) is approximately $N(\overline{Y}, 1/n)$. That is, $\sqrt{n}(\overline{Y}^* - \overline{Y}) \approx N(0,1)$. Let E^* denote the average with respect to the bootstrap randomness. Then, if $\widetilde{\theta}_n$ is the bagged estimator, we have

$$\begin{aligned} \widetilde{\theta}_n &= E^*[I(\overline{Y}^* \leq x_n)] = E^*\left[I\left(\sqrt{n}(\overline{Y}^* - \overline{Y}) \leq \sqrt{n}(x_n - \overline{Y}) \right) \right] \\ &= \Phi(\sqrt{n}(x_n - \overline{Y})) + o(1) = \Phi(c + Z) + o(1) \end{aligned}$$

where $Z \sim N(0,1)$, and we used the fact that $\overline{Y} \approx N(\mu, 1/n)$.

To summarize, $\widehat{\theta}_n \approx I(Z \leq c)$ while $\widetilde{\theta}_n \approx \Phi(c+Z)$ 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 $k \to \infty$ then*

$$\mathbb{E}[|\widehat{m}(X) - m(X)|^2] \to 0$$

as $n \to \infty$.

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* $t_n \to \infty$, $a_n \to \infty$ *and* $t_n(\log a_n)^9/a_n \to 0$ *then*

$$\mathbb{E}[|\widehat{m}(X) - m(X)|^2] \to 0$$

as $n \to \infty$.

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 $\widehat{m}(x)$ 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 $\widehat{m}(x)$ is some weighted average of the $k - PNN$'s.

Let us know 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

$$\widehat{m}(x) = \frac{1}{L_n(x)} \sum_i Y_i I(X_i \in \mathcal{L}_n(x)).$$

Biau and Devroye (2010) showed that, if X has a continuous density,

$$\frac{(d-1)!\mathbb{E}[L_n(x)]}{2^d(\log n)^{d-1}} \to 1.$$

Moreover, if Y is bounded and m is continuous then, for all $p \geq 1$,

$$\mathbb{E}|\widehat{m}_n(X) - m(X)|^p \to 0$$

as $n \to \infty$. Unfortunately, the rate of convergence is slow. Suppose that $\text{Var}(Y|X = x) = \sigma^2$ is constant. Then

$$\mathbb{E}|\widehat{m}_n(X) - m(X)|^p \geq \frac{\sigma^2}{\mathbb{E}[L_n(x)]} \sim \frac{\sigma^2(d-1)!}{2^d(\log n)^{d-1}}.$$

If we use k-PNN, with $k \to \infty$ 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 $k \to \infty$ and $k = o(n)$, then for all $p \geq 1$ and all distributions P, we have that $\mathbb{E}|\widehat{m}(X) - m(X)|^p \to 0$. 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

$$\widehat{m}(x) = \frac{1}{M}\sum_j \sum_i \frac{Y_i I(X_i \in A_j(x))}{N_j(x)} = \frac{1}{M}\sum_j \sum_i W_{ij} Y_j$$

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

$$\widehat{m}(x) = \frac{\sum_j \sum_i Y_i I(X_i \in A_j(x))}{\sum_j N_j(x)}.$$

With this modification, $\widehat{m}(x)$ is the average of each Y_i weighted by how often it appears in the trees. The KeRF can be written as

$$\widehat{m}(x) = \frac{\sum_i Y_i K(x, X_i)}{\sum_s K_n(x, X_s)}$$

where

$$K_n(x, z) = \frac{1}{M}\sum_j I(x \in A_j(x)).$$

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 $T(\Theta_1), \ldots, T(\Theta_M)$. And we can write $A_j(x)$ as $A(x, \Theta_j)$. Then $K_n(x, z)$ converges almost surely (as $M \to \infty$) 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

$$\mathbb{E}[(\widehat{m}(x) - m(x))^2] \leq C(\log n)^2 \left(\frac{1}{n}\right)^{\frac{1}{3+d\log 2}}.$$

This is slower than the minimax rate $n^{-2/(d+2)}$ but this probably reflects the difficulty in analyzing forests.

7 Variable Importance

Let $\widehat{m}$ 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 $\widehat{m}_{(-j)}$. Let $\mathcal{H}$ be a hold-out sample of size m. Then let

$$\widehat{\Delta}_j = \frac{1}{m}\sum_{i\in\mathcal{H}} W_i$$

where

$$W_i = (Y_i - \widehat{m}_{(-j)}(X_i))^2 - (Y_i - \widehat{m}(X_i))^2.$$

Then Δ_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,

$$\mathbb{E}[\widehat{\Delta}_j|\mathcal{T}] = \mathbb{E}\left[(Y - \widehat{m}_{(-j)}(X))^2 - (Y - \widehat{m}(X))^2 \;\middle|\; \mathcal{T}\right] \equiv \Delta_j.$$

In fact, since $\widehat{\Delta}_j$ 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 $N_{ij} = 1$ of (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 $X_1, \ldots, X_n \sim P$ where $X_i \in \mathbb{R}^d$. Let $C = \{c_1, \ldots, c_k\}$ where each $c_j \in \mathbb{R}^d$. We call C a codebook. Let $\Pi_C[X]$ be the projection of X onto C:

$$\Pi_C[X] = \text{argmin}_{c \in C} ||c - X||^2. \tag{1}$$

Define the empirical clustering risk of a codebook C by

$$R_n(C) = \frac{1}{n} \sum_{i=1}^{n} \Big|\Big|X_i - \Pi_C[X_i]\Big|\Big|^2 = \frac{1}{n} \sum_{i=1}^{n} \min_{1 \le j \le k} ||X_i - c_j||^2. \tag{2}$$

Let $\mathcal{C}_k$ denote all codebooks of length k. The optimal codebook $\widehat{C} = \{\widehat{c}_1, \ldots, \widehat{c}_k\} \in \mathcal{C}_k$ minimizes $R_n(C)$:

$$\widehat{C} = \text{argmin}_{C \in \mathcal{C}_k} R_n(C). \tag{3}$$

The empirical risk is an estimate of the population clustering risk defined by

$$R(C) = \mathbb{E}\Big|\Big|X - \Pi_C[X]\Big|\Big|^2 = \mathbb{E} \min_{1 \le j \le k} ||X - c_j||^2 \tag{4}$$

where $X \sim P$. The optimal population quantization $C^* = \{c_1^*, \ldots, c_k^*\} \in \mathcal{C}_k$ 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 $C = \{c_1, \ldots, c_k\}$ defines a set of cells known as a *Voronoi tesselation*. Let

$$V_j = \Big\{x : \ ||x - c_j|| \le ||x - c_s||, \ \text{ for all } s \ne j\Big\}. \tag{5}$$

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.*

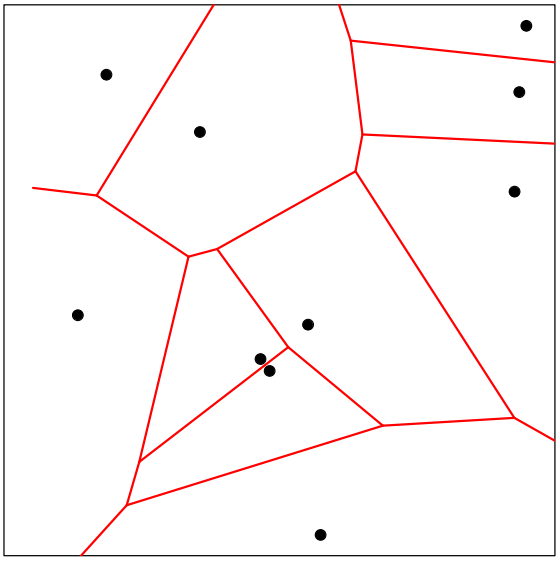


Figure 1: *The Voronoi tesselation formed by 10 cluster centers* $c_1, \ldots, c_{10}$*. The cluster centers are indicated by dots. The corresponding Voronoi cells* $T_1, \ldots, T_{10}$ *are defined as follows: a point* x *is in* T_j *if* x *is closer to* c_j *than* c_i *for* $i \neq j$*.*

1. Choose k centers $c_1, \ldots, c_k$ as starting values.
2. Form the clusters $C_1, \ldots, C_k$ as follows. Let $g = (g_1, \ldots, g_n)$ where $g_i = \text{argmin}_j ||X_i - c_j||$. Then $C_j = \{X_i : \ g_i = j\}$.
3. For $j = 1, \ldots, k$, let n_j denote the number of points in C_j and set

$$c_j \longleftarrow \frac{1}{n_j} \sum_{i: \ X_i \in C_j} X_i.$$

4. Repeat steps 2 and 3 until convergence.
5. Output: centers $\widehat{C} = \{c_1, \ldots, c_k\}$ and clusters $C_1, \ldots, C_k$.

Figure 2: The k-means (Lloyd's) clustering algorithm.

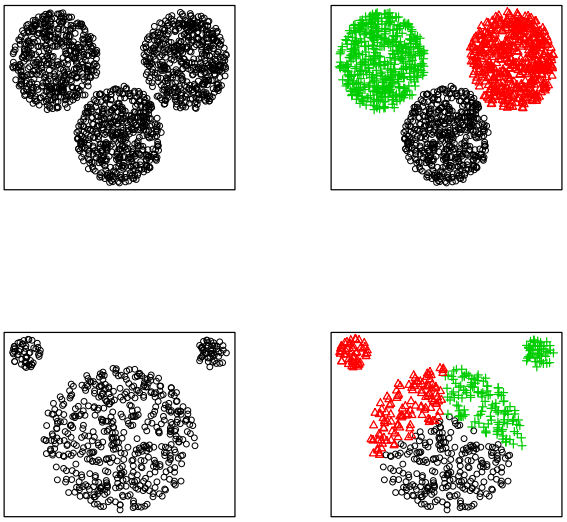


Figure 3: *Synthetic data inspired by the "Mickey Mouse" example from wikipedia. Top left: three balanced clusters. Top right: result from running k means with $k = 3$. Bottom left: three unbalanced clusters. Bottom right: result from running k means with $k = 3$ on the unbalanced clusters. k-means does 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.

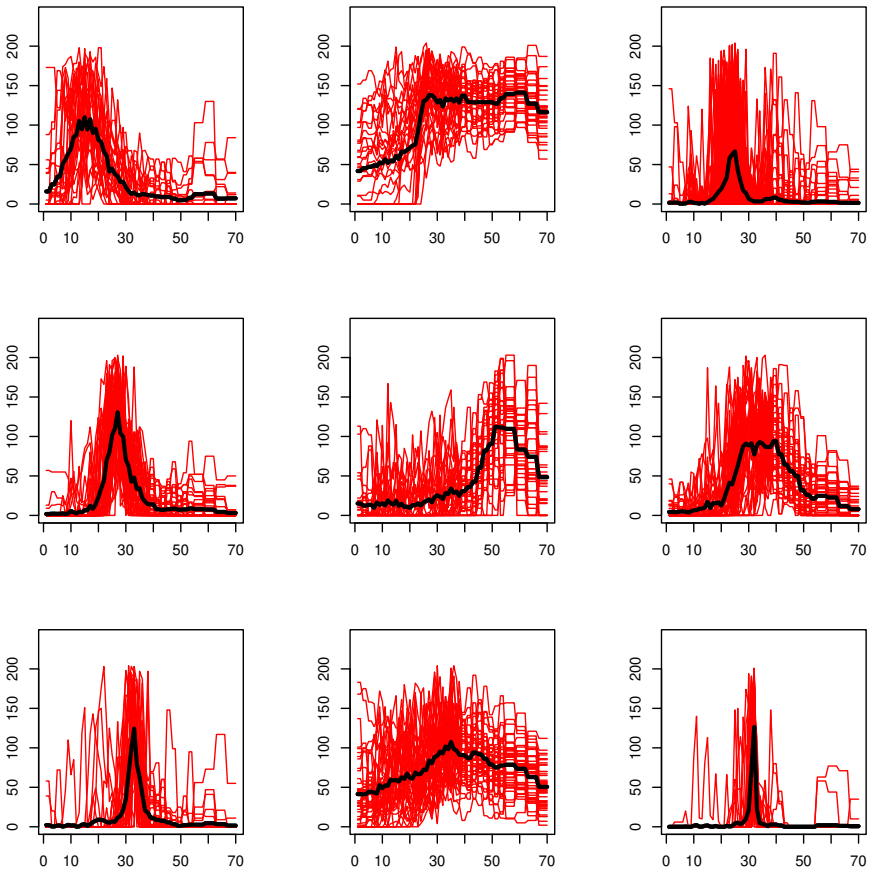


Figure 4: The nine clusters found in the Topex data using k-means clustering with $k = 9$. Each plot show the curves in that cluster together with the mean of the curves in that cluster.

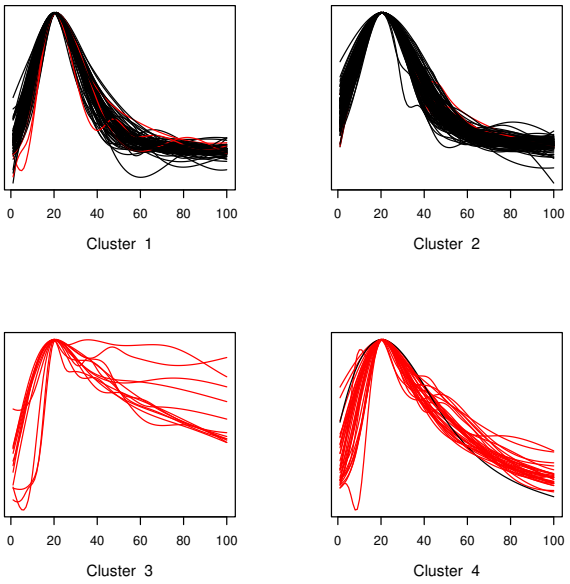


Figure 5: Clustering of the supernova light curves with $k = 4$.

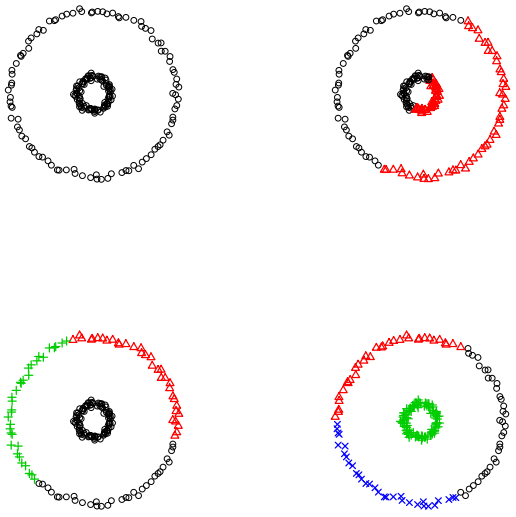


Figure 6: Top left: a dataset with two ring-shaped clusters. Top right: k-means with $k = 2$. Bottom left: k-means with $k = 3$. Bottom right: k-means with $k = 4$.

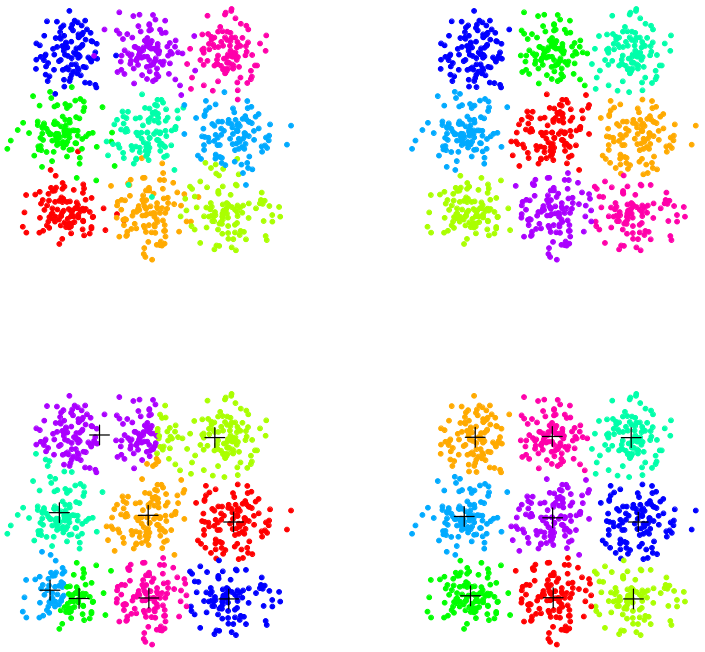


Figure 7: An example with 9 clusters. Top left: data. Top right: k-means with random starting values. Bottom left: k-means using starting values from hierarchical clustering. Bottom right: the k-means^{++} algorithm.

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 staring values for k-means. Run single-linkage hierarchical clustering (which we describe in Section 6) to obtains $p \times k$ 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* $C = \{c_1, \ldots, c_k\}$ *be the seed points from*

1. Input: Data $X = \{X_1, \ldots, X_n\}$ and an integer k.
2. Choose c_1 randomly from $X = \{X_1, \ldots, X_n\}$. Let $C = \{c_1\}$.
3. For $j = 2, \ldots, k$:
   (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 $C \longleftarrow C \cup \{c_j\}$.
4. Run Lloyd's algorithm using the **seed points** $C = \{c_1, \ldots, c_k\}$ as starting points and output the result.

Figure 8: The k-means^{++} algorithm.

the k-means^{++} algorithm. Then,

$$\mathbb{E}\big(R_n(C)\big) \leq 8(\log k + 2) \ \Big(\min_C R_n(C)\Big) \tag{6}$$

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 $\widehat{R}_k$ 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 $\widehat{R}_k$. We can try to make this more formal by fixing a small number $\alpha > 0$ and defining

$$k_\alpha = \min\left\{k : \ \frac{R_k - R_{k+1}}{\sigma^2} \leq \alpha\right\} \tag{7}$$

where $\sigma^2 = \mathbb{E}(\|X - \mu\|^2)$ and $\mu = \mathbb{E}(X)$. An estimate of k_α is

$$\widehat{k}_\alpha = \min\left\{k : \ \frac{\widehat{R}_k - \widehat{R}_{k+1}}{\widehat{\sigma}^2} \leq \alpha\right\} \tag{8}$$

where $\widehat{\sigma}^2 = n^{-1}\sum_{i=1}^n \|X_i - \overline{X}\|^2$.

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, $\widehat{k}$ 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. The 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* $k = 3$*.*

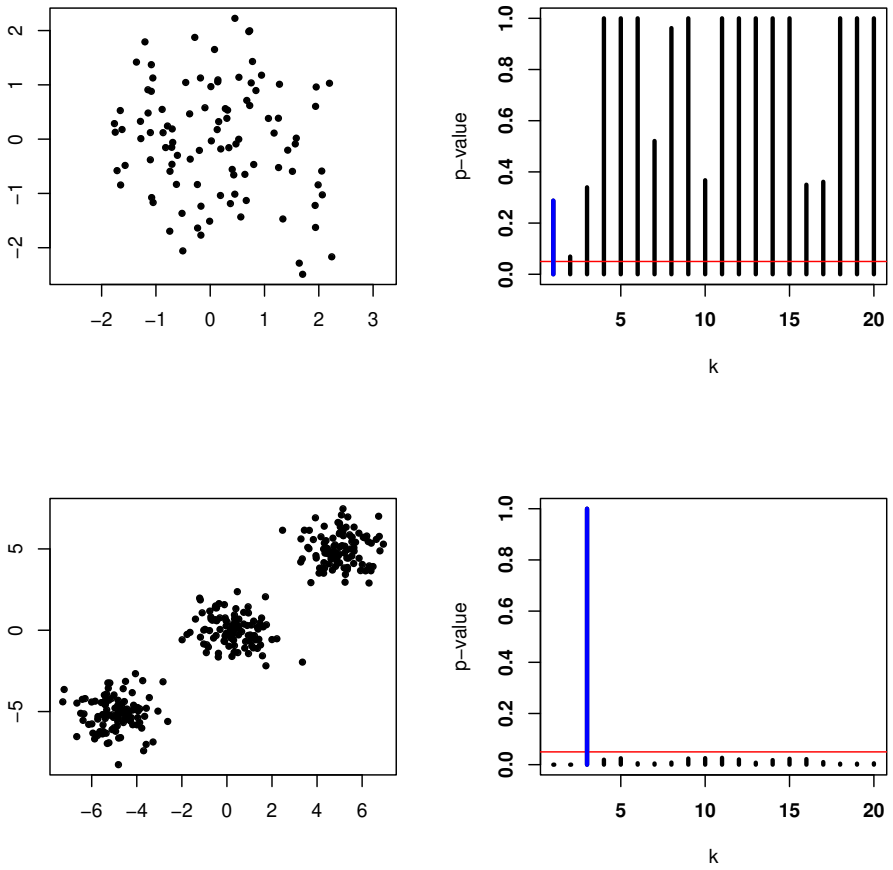


Figure 9: Top left: a single cluster. Top right: p-values for various k. The first k for which the p-value is larger than .05 is $k = 1$. Bottom left: three clusters. Bottom right: p-values for various k. The first k for which the p-value is larger than .05 is $k = 3$.

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 $Y = (Y_1, \ldots, Y_n)$ and $Z = (Z_1, \ldots, Z_n)$ 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*

$$\Omega(k) = \mathbb{E}\left[s(A_k(Y), A_k(Z))\right] \tag{9}$$

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 $X = (X_1, \ldots, X_{2n})$. 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

$$\widehat{\Omega}(k) = \frac{1}{N}\sum_{j=1}^{N}\left[s(A_k(Y^j), A_k(Z^j))\right].$$

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 $\{1, \ldots, k\}$ 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

$$\psi_{Y,Z}(Z_j, Z_k) = \psi_Y(Y'_j, Y'_k) \tag{10}$$

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

$$s(\psi_{Y,Z}, \psi_Z) = \frac{1}{\binom{n}{2}} \sum_{s \neq t} I\left(\psi_{Y,Z}(Z_s, Z_t) = \psi_Z(Z_s, Z_t)\right).$$

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 Rakhlin (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 $C^* = \{c_1^*, \ldots, c_k^*\}$ minimizes $R(C) = \mathbb{E}||X - \Pi_C[X]\,||^2$.

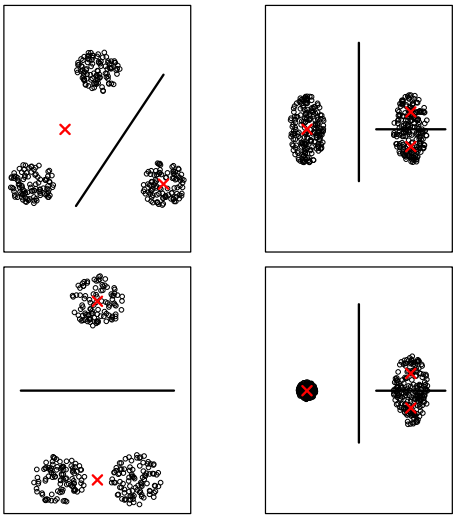


Figure 10: Examples from Ben-David, von Luxburg and Pál (2006). The first example (top left plot) shows a case where we fit $k = 2$ clusters. Stability analysis will correctly show that k is too small. The top right plot has $k = 3$. Stability analysis will correctly show that k is too large. The bottom 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*

$$\mathbb{E}(R(\widehat{C})) - R(C^*) \leq c\sqrt{\frac{k(d+1)\log n}{n}} \tag{11}$$

for some $c > 0$.

Warning! The fact that $R(\widehat{C})$ is close to $R(C_*)$ does not imply that $\widehat{C}$ 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 $R(\widehat{C}) - R(C^*) = R(\widehat{C}) - R_n(\widehat{C}) + R_n(\widehat{C}) - R(C^*) \leq R(\widehat{C}) - R_n(\widehat{C}) + R_n(C^*) - R(C^*) \leq 2\sup_{C\in\mathcal{C}_k} |R(\widehat{C}) - R_n(\widehat{C})|$. 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 $\psi = (\pi_0, \ldots, \pi_{K-1}, \theta_0, \ldots, \theta_{K-1})$ are the unknown parameters. We call $p_{\theta_0}, \ldots, p_{\theta_{K-1}}$ 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

$$K\Bigg(\underbrace{\frac{d(d+1)}{2}}_{\text{\# of parameters in } \Sigma_j} + \underbrace{d}_{\text{\# of parameters in } u_j}\Bigg) + \underbrace{(K-1)}_{\text{\# of mixing coefficients}} = \frac{Kd(d+3)}{2} + K - 1$$

parameters in the mixture of K multivariate Gausssians.

3.2 Maximum Likelihood Estimation

A finite mixture model $p_\psi(x)$ has parameters $\psi = (\pi_0, \ldots, \pi_{K-1}, \theta_0, \ldots, \theta_{K-1})$. The likelihood of ψ based on the observations $X_1, \ldots, X_n$ is

$$\mathcal{L}(\psi) = \prod_{i=1}^{n} p_\psi(X_i) = \prod_{i=1}^{n}\Bigg(\sum_{j=0}^{K-1} \pi_j p_{\theta_j}(X_i)\Bigg)$$

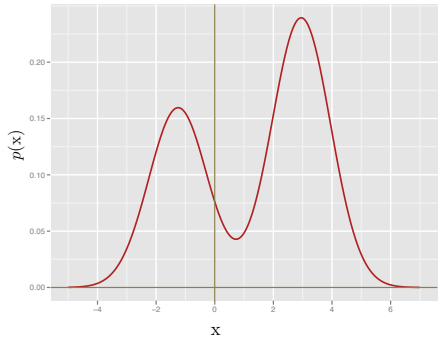


Figure 11: A mixture of two Gaussians, $p(x) = \frac{2}{5}\phi(x; -1.25, 1) + \frac{3}{5}\phi(x; 2.95, 1)$.

and, as usual, the maximum likelihood estimator is the value $\widehat{\psi}$ that maximizes $\mathcal{L}(\psi)$. Usually, the likelihood is multimodal and one seeks a local maximum instead if a global maximum.

For fixed $\theta_0, \ldots, \theta_{K-1}$, the log-likelihood is often a concave function of the mixing parameters π_j. However, for fixed $\pi_0, \ldots, \pi_{K-1}$, the log-likelihood is not generally concave with respect to $\theta_0, \ldots, \theta_{K-1}$.

One way to find $\widehat{\psi}$ 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.

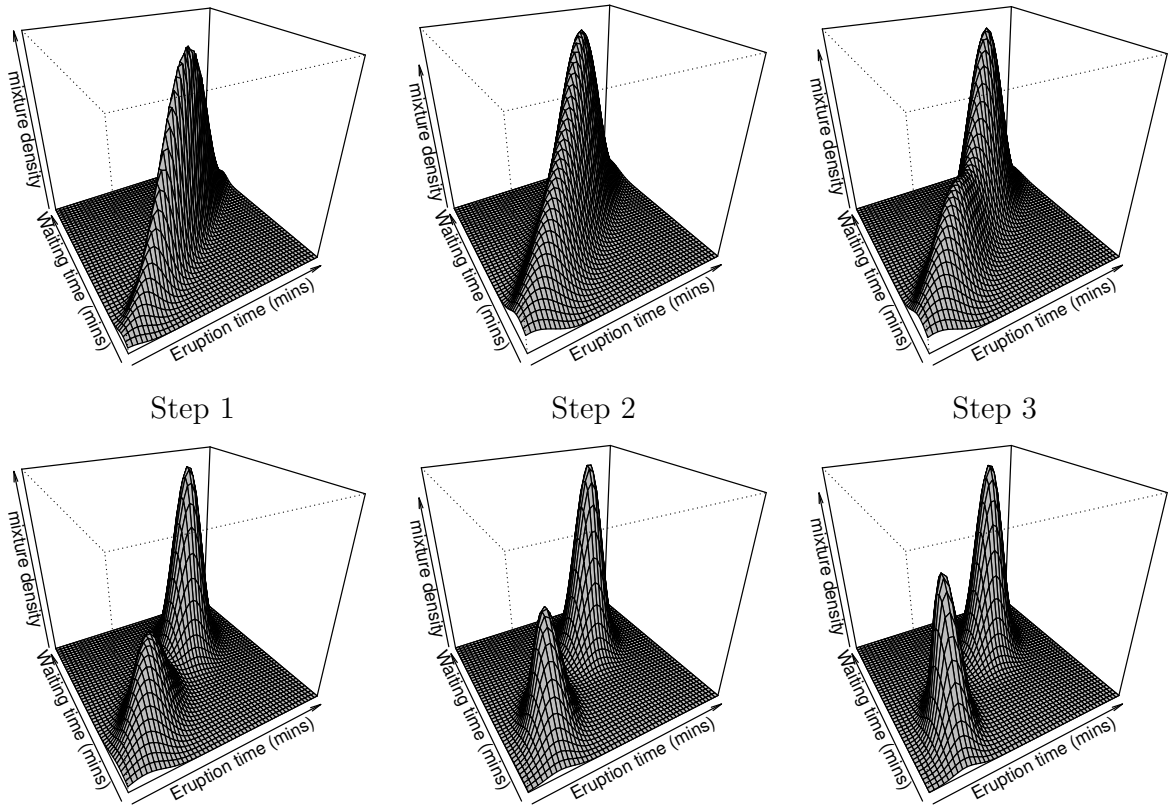


Figure 12: Fitting a mixture of two Gaussians on the Old Faithful Geyser data. The initial values are $\pi_0 = \pi_1 = 0.5$. $u_0 = (4, 70)^T$, $u_1 = (3, 60)^T$, $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 0.8 & 7 \\ 7 & 70 \end{pmatrix}$. We see that even though the starting density is not bimodal, the 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 $\sigma_j \to 0$. We have $\phi(X_1; \mu_j, \sigma_j^2) \to \infty$. 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.

Nonidentifability. A model $\{p_\theta(x) : \theta \in \Theta\}$ is identifiable if

$$\theta_1 \neq \theta_2 \quad \text{implies} \quad P_{\theta_1} \neq P_{\theta_2}$$

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

$$p(x; \pi, \mu_1, \mu_2) = (1-\pi)\phi(x; \mu_1, 1) + \pi\phi(x; \mu_2, 1). \tag{13}$$

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 identifiabilty causes other problems too. For example, we usually have that the Fisher information is non-zero and that $\widehat{\theta} - \theta = O_P(n^{-1/2})$ where $\widehat{\theta}$ 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.

Nonintinuitive 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 $p(\psi|X_1, \ldots, X_n) \propto \mathcal{L}(\psi)\pi(\psi)$. Here, $\pi(\psi)$ is the prior distribution that represents our knowledge of ψ before

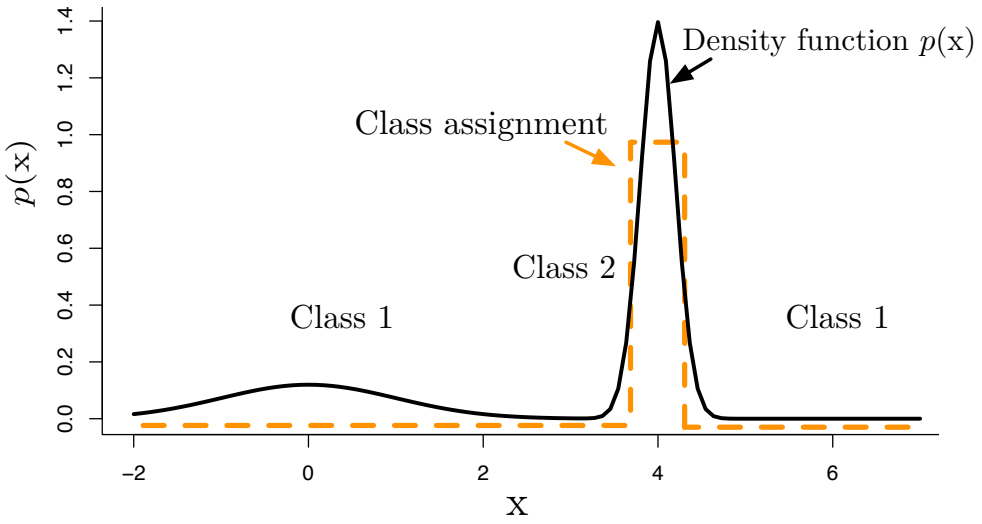


Figure 13: Mixtures are used as a parametric method for finding clusters. Observations with $x = 0$ and $x = 6$ are both classified into the first component.

seeing the data. Often, the prior is improper, meaning that it does not have a finite integral. For example, suppose that $X_1, \ldots, X_n \sim N(\mu, 1)$. 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 $N(\overline{X}, 1/\sqrt{n})$ where $\overline{x}$ 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

$$p(x; \mu) = \frac{1}{2}\phi(x; 0, 1) + \frac{1}{2}\phi(x; \mu, 1). \tag{14}$$

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 modes. 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 modes. 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: $L_t = C_1 \bigcup \cdots \bigcup C_{k_t}$. We call $\mathcal{C}_t = \{C_1, \ldots, C_{k_t}\}$ 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 (2104) show that we can take $\epsilon = h$. G_t is a 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.brianpkent.com/projects.html`.

Fabrizio Lecci has written an R implementation, include 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.

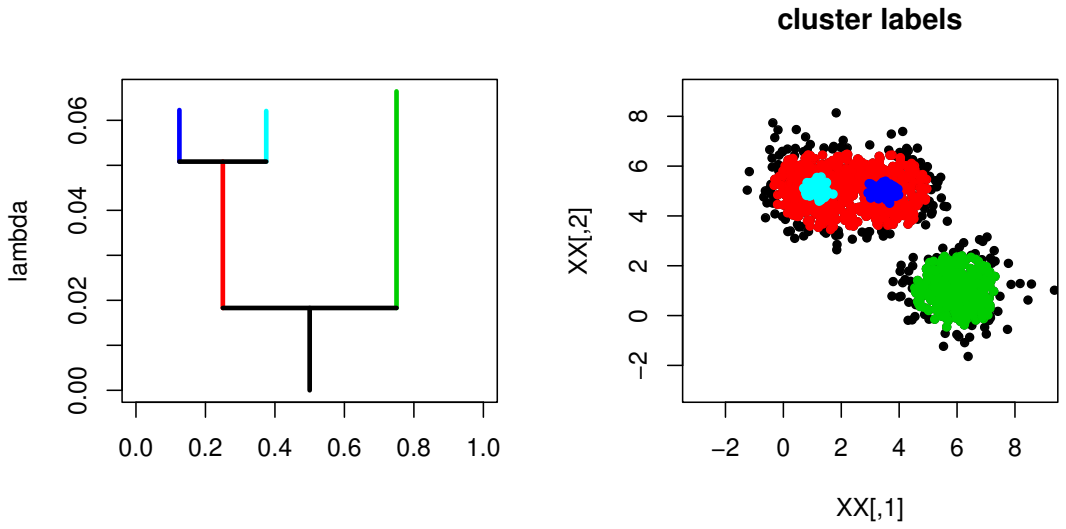


Figure 14: DeBaClR in two dimensions.

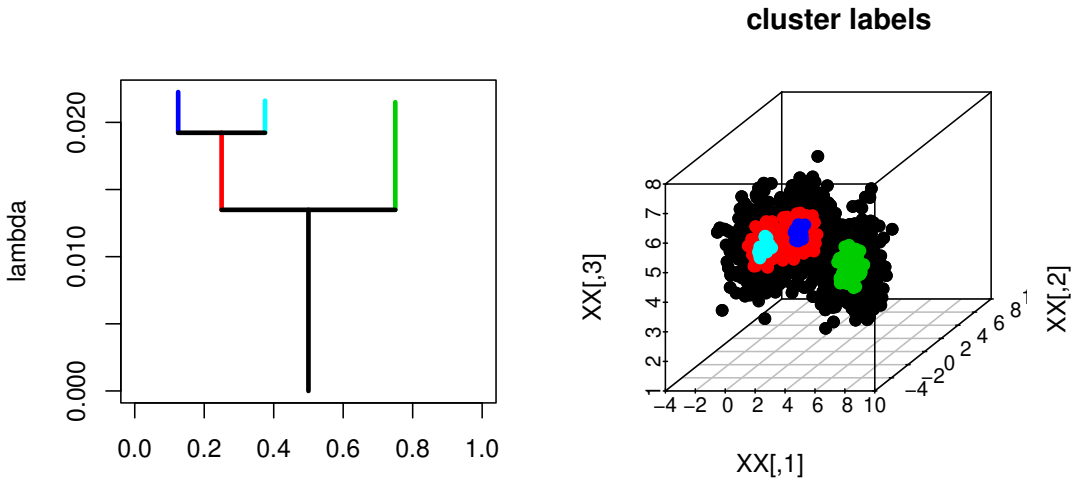


Figure 15: DeBaClR in three dimensions.

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 $||x-y|| \leq Cr_n$. □

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.

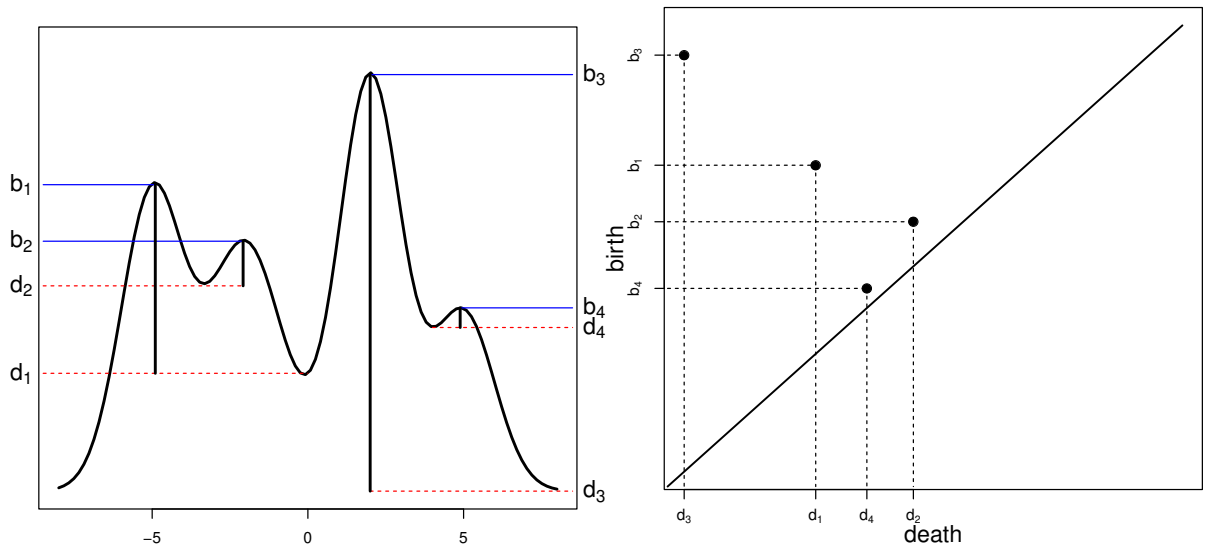


Figure 16: *Starting at the top of the density and moving down, each mode has a birth time b and a death time d. The persistence diagram (right) plots the points $(d_1, b_1), \ldots, (d_4, b_4)$. Modes with a long lifetime are far from the diagonal.*

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 $(d_1, b_1), \ldots, (d_k, b_k)$ 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

$$\epsilon_\alpha = \inf\left\{ z : \ \frac{1}{B}\sum_{b=1}^{B} I\Big(||\widehat{p}_h^{*b} - \widehat{p}_h||_\infty > z\Big) \leq \alpha \right\}. \tag{15}$$

Here, $\widehat{p}_h^{*b}$ 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 $m_1, \ldots, m_{k_0}$ and that p is a *Morse function*, which means that the Hessian of p at each stationary point is non-degenerate. We

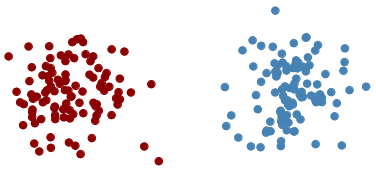


Figure 17: A synthetic example with two "blob-like" clusters.

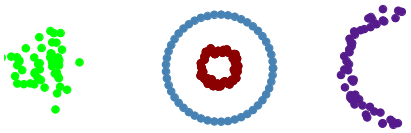


Figure 18: A synthetic example with four clusters with a variety of different shapes.

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 $\pi_x : \mathbb{R} \to \mathbb{R}^d$ such that $\pi_x(0) = x$ and

$$\pi_x'(t) = \nabla p(\pi_x(t)). \tag{16}$$

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

$$\text{dest}(x) = \lim_{t \to \infty} \pi_x(t). \tag{17}$$

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

$$\mathcal{A}_j = \Big\{ x : \ \text{dest}(x) = m_j \Big\}. \tag{18}$$

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*.

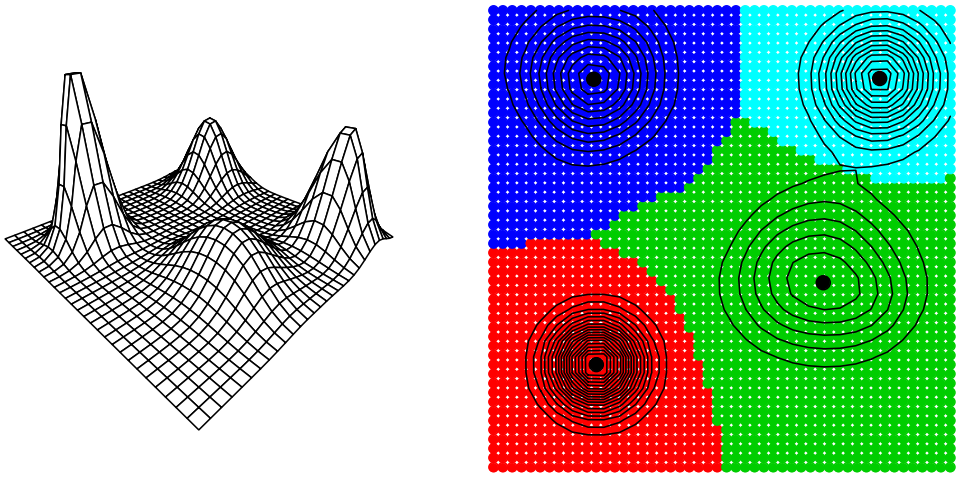


Figure 19: *The left plot shows a function with four modes. The right plot shows the ascending manifolds (basins of attraction) corresponding to the four modes.*

Given data $X_1, \ldots, X_n$ we construct an estimate $\widehat{p}$ of the density. Let $\widehat{m}_1, \ldots, \widehat{m}_k$ be the estimated modes and let $\widehat{\mathcal{A}}_1, \ldots, \widehat{\mathcal{A}}_k$ be the corresponding ascending manifolds derived from $\widehat{p}$. The sample clusters $C_1, \ldots, C_k$ are defined to be $C_j = \left\{X_i : \ X_i \in \widehat{\mathcal{A}}_j\right\}$.

Recall that the kernel density estimator is

$$\widehat{p}(x) \equiv \widehat{p}_h(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^d} K\left(\frac{||x - X_i||}{h}\right) \tag{19}$$

where K is a smooth, symmetric kernel and $h > 0$ is the bandwidth.[1] The mean of the estimator is

$$p_h(x) = \mathbb{E}[\widehat{p}_h(x)] = \int K(t)p(x + th)dt. \tag{20}$$

To locate the modes of $\widehat{p}_h$ 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 $\widehat{\mathcal{M}} = \{\widehat{m}_1, \ldots, \widehat{m}_k\}$. 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.

[1] In general, we can use a bandwidth matrix H in the estimator, with $\widehat{p}(x) \equiv \widehat{p}_H(x) = \frac{1}{n}\sum_{i=1}^{n} K_H(x - X_i)$ where $K_H(x) = |H|^{-\frac{1}{2}} K(H^{-\frac{1}{2}}x)$.

Mean Shift Algorithm

1. Input: $\widehat{p}(x)$ and a mesh of points $A = \{a_1, \ldots, a_N\}$ (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:

$$a_j^{(s+1)} \longleftarrow \frac{\sum_{i=1}^n X_i K\left(\frac{||a_j^{(s)} - X_i||}{h}\right)}{\sum_{i=1}^n K\left(\frac{||a_j^{(s)} - X_i||}{h}\right)}.$$

3. Let $\widehat{\mathcal{M}}$ be the unique values of the set $\{a_1^{(\infty)}, \ldots, a_N^{(\infty)}\}$.
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 $\phi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ 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.

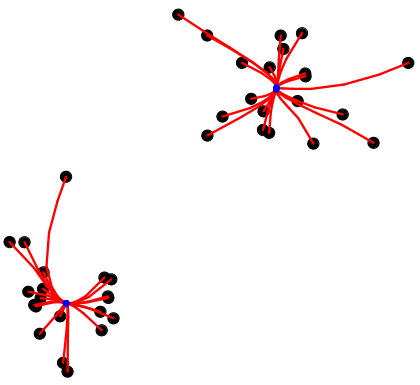


Figure 21: A simple example of the mean shift algorithm.

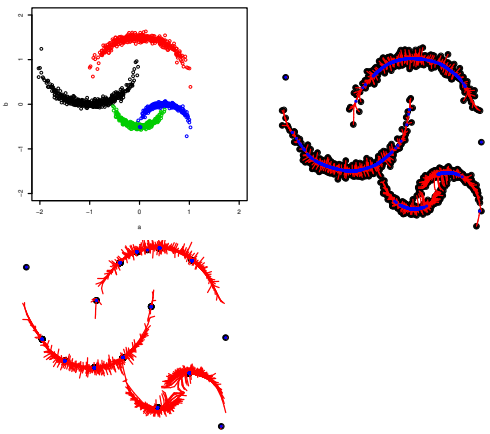


Figure 22: The crescent data example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

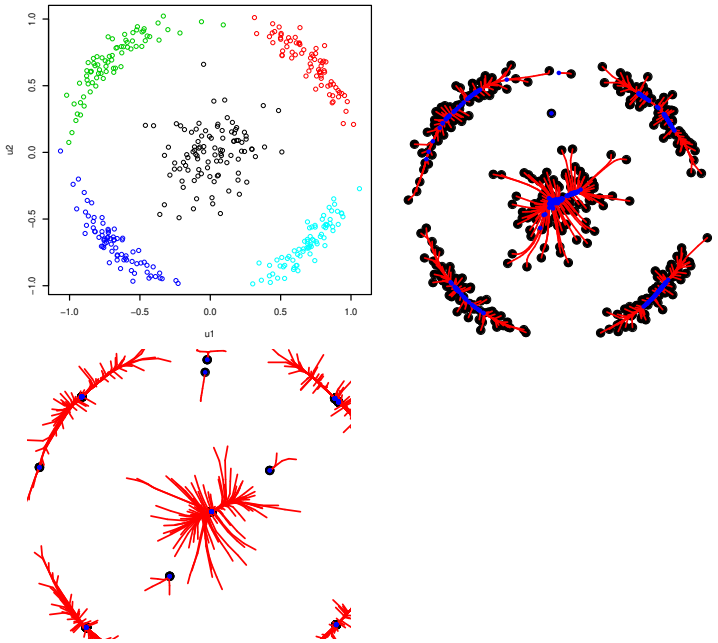


Figure 23: The Broken Ring example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

Theorem 12 (Silverman 1981) *Let $\widehat{p}_h$ be a kernel density estimator using a Gaussian kernel in one dimension. Then the number of modes of $\widehat{p}_h$ 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

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

We can ignore the last term. The first term is known. To estimate the middle term, we use integration by parts to get

$$\int \widehat{p}p' = -\int p''p$$

suggesting the cross-validation estimator

$$\int (\widehat{p}'(x))^2 dx + \frac{2}{n}\sum_i \widehat{p}_i''(X_i)$$

where $\widehat{p}_i''$ 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

$$\text{CV}_r(h) = \int (\widehat{p}^{(r)}(x))^2 dx - \frac{2}{n}(-1)^r \sum_i \widehat{p}_i^{(2r)}(X_i).$$

Let's now discuss estimating derivatives more generally following Chacon and Duong (2013). Let

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

where $K_H(x) = |H|^{-1/2}K(H^{-1/2}x)$. Let $D = \partial/\partial x = (\partial/\partial x_1, \ldots, \partial/\partial x_d)$ 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

$$D^{\otimes 0}p = p, \qquad D^{\otimes 1}p = Dp, \qquad D^{\otimes 2}p = \text{vec}(H)$$

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

$$\begin{aligned}\mathbb{E}[L] = \frac{1}{n}|H|^{-1/2}\mathrm{tr}((H^{-1})^{\otimes r}R(D^{\otimes r}K)) - \frac{1}{n}\mathrm{tr}R^*(K_H \star K_H, D^{\otimes r}p) \\ + \mathrm{tr}R^*(K_H \star K_H, D^{\otimes r}p) - 2\mathrm{tr}R^*(K_H, D^{\otimes r}p) + \mathrm{tr}R(D^{\otimes r}p)\end{aligned}$$

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:

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

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 $m_1, \ldots, m_k$. Then for $h > 0$ and not too large, p_h is Morse with modes $m_{h1}, \ldots, m_{hk}$ and (possibly after relabelling),*

$$\max_j ||m_j - m_{jh}|| = O(h^2).$$

With probability tending to 1, $\widehat{p}_h$ has the same number of modes which we denote by $\widehat{m}_{h1}, \ldots, \widehat{m}_{hk}$. Furthermore,

$$\max_j ||\widehat{m}_{jh} - m_{jh}|| = O_P\left(\sqrt{\frac{1}{nh^{d+2}}}\right)$$

and

$$\max_j ||\widehat{m}_{jh} - m_j|| = O(h^2) + O_P\left(\sqrt{\frac{1}{nh^{d+2}}}\right).$$

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 if 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

$$\max_j ||\widehat{m}_{jh} - m_{jh}|| = O_P\left(\sqrt{\frac{1}{nh^{d+2}}}\right).$$

For simplicity, write $m = m_{jh}$ and $x = \widehat{m}_{jh}$. Let $g(x)$ and $H(x)$ be the gradient and Hessian of $p_h(x)$ and let $\widehat{g}(x)$ and $\widehat{H}(x)$ be the gradient Hessian of $\widehat{p}_h(x)$. Then

$$(0, \ldots, 0)^T = \widehat{g}(x) = \widehat{g}(m) + (x - m)^T \int_0^1 \widehat{H}(m + u(x - m))du$$

and so

$$(x - m)^T \int_0^1 \widehat{H}(m + u(x - m))du = (g(m) - \widehat{g}(m))$$

where we used the fact that $\mathbf{0} = g(m)$. Multiplying on the right by $x - m$ we have

$$(x - m)^T \int_0^1 \widehat{H}(m + u(x - m))(x - m)du = (\widehat{g}(m) - \widehat{g}(m))^T(x - m).$$

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 $||x-m|| = O_P\left(\sqrt{\frac{1}{nh^{d+2}}}\right)$. □

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. The 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 to 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 $X = \{X_1, \ldots, X_n\}$ and metric d giving distance between clusters.
2. Let $T_n = \{C_1, C_2, \ldots, C_n\}$ where $C_i = \{X_i\}$.
3. For $j = n-1$ to 1:

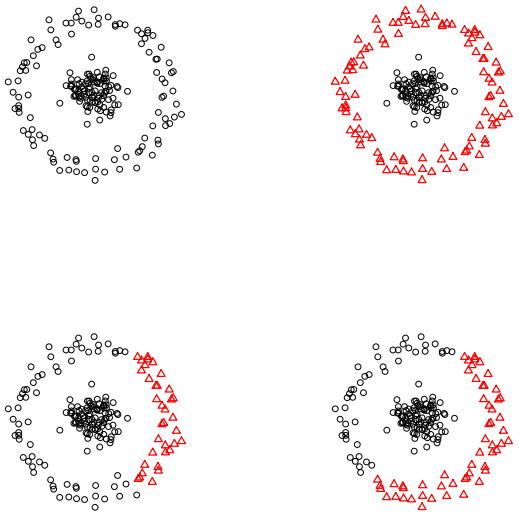


Figure 24: Hierarchical clustering applied to two noisy rings. Top left: the data. Top right: two clusters from hierarchical clustering using single linkage. Bottom left: average linkage. Bottom right: complete linkage.

   (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 $T_1, \ldots, T_n$.

The result can be represented as a tree, called a dendogram. 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 show 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 $X_1, \ldots, X_n$ 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 $V = \{X_1, \ldots, X_n\}$ and $E_{ij} = 1$ if $||X_i - X_j|| \leq \epsilon$ and $E_{ij} = 0$ if $||X_i - X_j|| > \epsilon$. Let $C_1, \ldots, C_k$ 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 $X_1, \ldots, X_n$. 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* $f = (f_1, \ldots, f_n)^T$,
$$f^T L f = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}(f_i - f_j)^2.$$
2. *L is symmetric and positive semi-definite.*
3. *The smallest eigenvalue of L is 0. The corresponding eigenvector is* $(1, 1, \ldots, 1)^T$.
4. *L has n non-negative, real-valued eigenvalues* $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k$.
5. *The number of eigenvalues that are equal to 0 is equal to the number of connected components of G. That is, $0 = \lambda_1 = \ldots = \lambda_k$ where k is the number of connected components of G. The corresponding eigenvectors $v_1, \ldots, v_k$ 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 follow that L is symmetric. The fact that L is positive semi-definite folows from part (1).

(3) Let $v = (1, \ldots, 1)^T$. 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 $v_1 = (1, \ldots, 1)^T$. Suppose there were another eigenvector v with eigenvalue 0. Then
$$0 = v^T L v = \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}(v(i) - v(j))^2.$$

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 components j. We can reliable 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 $v_1 = (1, \ldots, 1, 0, \ldots, 0)$ where the 1's correspond to the first component. Let Let $v_2 = (0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0)$ where the 1's correspond to the second component. Define $v_3, \ldots, v_k$ 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 th connected components of the graph. It is easy to see that $LV - j = 0$ for $j = 1, \ldots, k$. Thus, each v_j, for $j = 1, \ldots, k$ 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 $1 \leq j \leq k$. □

Example 16 *Consider the graph*

and suppose that $W_{ij} = 1$ if and only if there is an edge between X_i and X_j. Then

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

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.

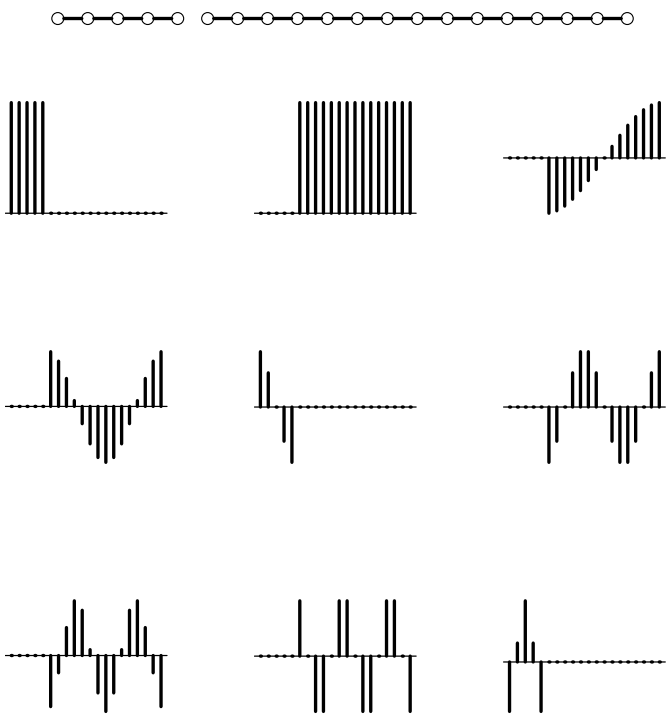


Figure 25: The top shows a simple graph. The remaining plots are the eigenvectors of the graph Laplacian. Note that the first two eigenvectors correspond to the two connected components of the graph.

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 two the connected components of the graph. The other eignvectors can be thought of as forming bases 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 $(1, \ldots, 1)^T$.*
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 $v_1, \ldots, v_k$ denote the eigenvectors with eigenvalues equal to 1. The linear space spanned by $v_1, \ldots, v_k$ is spanned by the indicator functions of the connected components.*

Proof. Homework. □ **H**

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of $\mathcal{L}$ with eigenvectors $v_1, \ldots, v_n$. Define

$$Z_i \equiv T(X_i) = \sum_{j=1}^{r} \sqrt{\lambda_j}\, v_j(i).$$

The mapping $T : X \to Z$ 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 $X_1, \ldots, X_n$ where we jump from X_i to X_j with probability

$$\mathbb{P}(X_i \longrightarrow X_j) = \mathcal{L}(i,j) = \frac{K_h(X_i, X_j)}{\sum_s K_h(X_s, X_j)}.$$

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:

$$P(x \to A) = \frac{\int_A K_h(x,y)dP(y)}{\int K_h(x,y)dP(y)}.$$

The corresponding averaging operator $\widehat{A} : f \to \widetilde{f}$ 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 $A : f \to \widetilde{f}$ 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 $v_1, \ldots, v_k$ of $\mathcal{L}$.
4. Project each X_i onto the eigenvectors to get new points $\widehat{X}_i$.
5. Cluster the points $\widehat{X}_1, \ldots, \widehat{X}_n$ 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.*

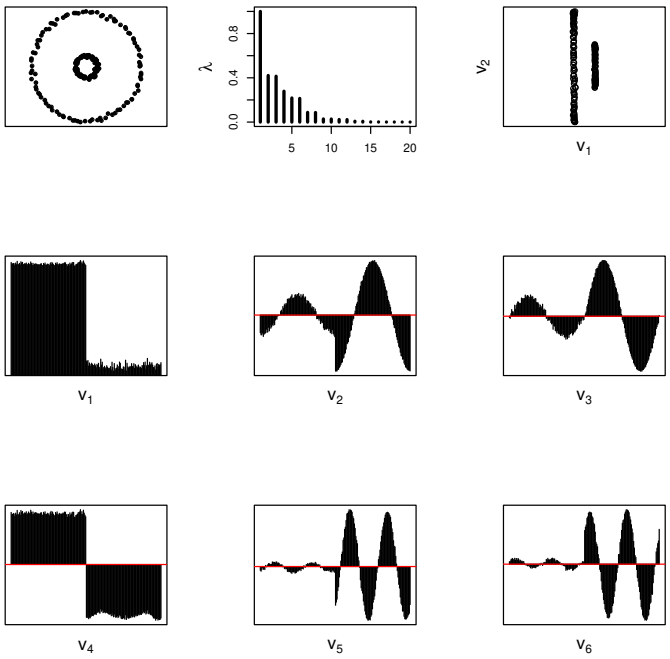


Figure 26: Top left: data. Top middle: eigenvalues. Top right: second versus third eigenvectors. Remaining plots: first six eigenvectors.

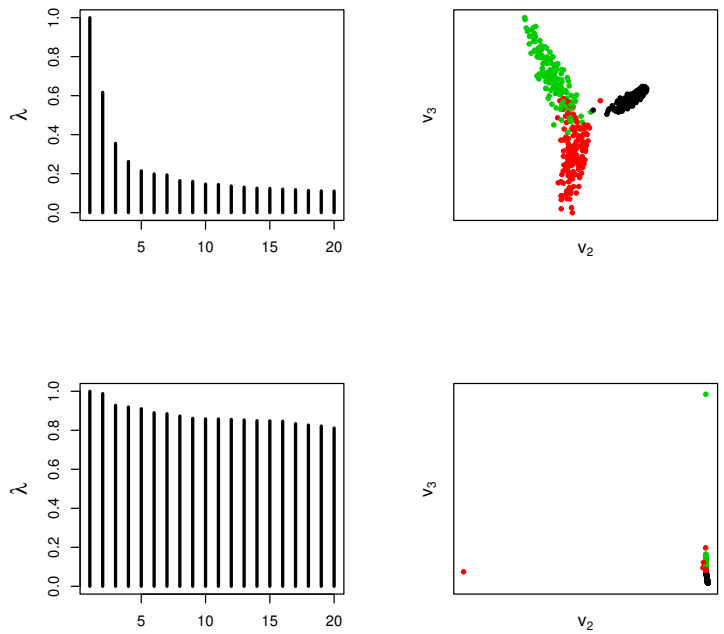


Figure 27: Spectral analysis of some zipcode data. Top: $h = 6$. Bottom: $h = 4$. The plots on the right show the second versus third eigenvector. The three colors correspond to the three digits 1, 2 and 3.

Example 20 *Figure 27 shows a spectral analysis of some zipcode data. Each datapoint is a 16 x 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 x 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 eignvectors 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 $P_1, \ldots, P_k$. 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

$$\nu_{rs} = \lim_{d\to\infty} \frac{1}{d} \sum_{a=1}^{d} ||\mu_r(a) - \mu_s(a)||^2$$

and

$$\sigma_r^2 = \lim_{d\to\infty} \frac{1}{d} \text{trace}(\Sigma_r).$$

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

$$\rho(x,y) = \frac{1}{n-2}\sum_{z\neq x,y}\Big|\,||x-z|| - ||y-z||\,\Big|.$$

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 $\widetilde{\alpha} = \alpha/(nd)$.
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 $T_j > c_{n,\widetilde{\alpha}}$ where $c_{n,\widetilde{\alpha}}$ 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 $Z_1, \ldots, Z_n \in [0,1]$ 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

$$\text{Dip}(F) = \inf_{G \in \mathcal{U}} \sup_x |F(x) - G(x)|. \tag{22}$$

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 $\widetilde{\alpha} = \alpha/(nd)$. That is, we test each marginal and we reject H_0 if $T_n > c_{n,\widetilde{\alpha}}$. By the union bound, the chance of at least one false rejection of H_0 is at most $d\widetilde{\alpha} = \alpha/n$.

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 $C = \{c_1, \ldots, c_k\}$ to minimize

$$R_n(C) = \frac{1}{n}\sum_{i=1}^n ||X_i - \Pi_C[X_i]||^2 = \frac{1}{n}\sum_{i=1}^n \min_{1 \leq j \leq k} ||X_i - c_j||^2. \tag{23}$$

This is equivalent to minimizing the within sums of squares

$$\sum_{j=1}^k \frac{1}{n_j} \sum_{s,t \in A_j} d^2(X_s, X_t) \tag{24}$$

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

$$B = \frac{1}{n}\sum_{s,t} d^2(X_s, X_t) - \sum_{j=1}^k \frac{1}{n_j} \sum_{s,t \in A_j} d^2(X_s, X_t). \tag{25}$$

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

$$B = \frac{1}{n}\sum_{s,t} d_w^2(X_s, X_t) - \sum_{j=1}^k \frac{1}{n_j} \sum_{s,t \in A_j} d_w^2(X_s, X_t) \tag{26}$$

[2]Specifically, $c_{n,\alpha}$ 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 Unif(0,1). Hartigan and Hartigan (1985) suggest that this suffices asymptotically.

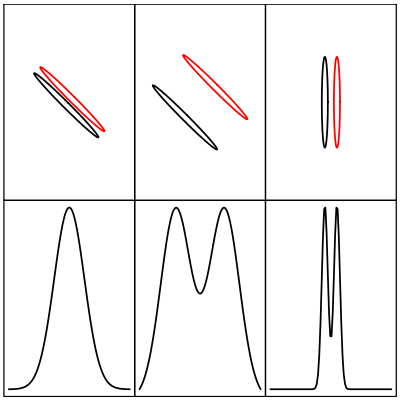


Figure 28: *Three examples, each showing two clusters and two features $X(1)$ and $X(2)$. The top plots show the clusters. The bottom plots show the marginal density of $X(1)$. Left: The marginal fails to reveal any clustering structure. This example violates the marginal signature assumption. Middle: 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 $w = (w_1, \ldots, w_d)$. 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 $X_1, \ldots, X_n$ and k.
2. Set $w = (w_1, \ldots, w_d)$ where $w_1 = \ldots = w_d = 1/\sqrt{d}$.
3. Iterate until convergence:
   (a) Optimize (25) over C holding w fixed. Find $c_1, \ldots, c_k$ 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 $c_1, \ldots, c_k$ 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 maginal 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 $X_1, \ldots, X_n$ 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 $f_1, \ldots, f_n$ and the goal is to cluster similarly shaped clusters together. For example, Figure 30 shows a*

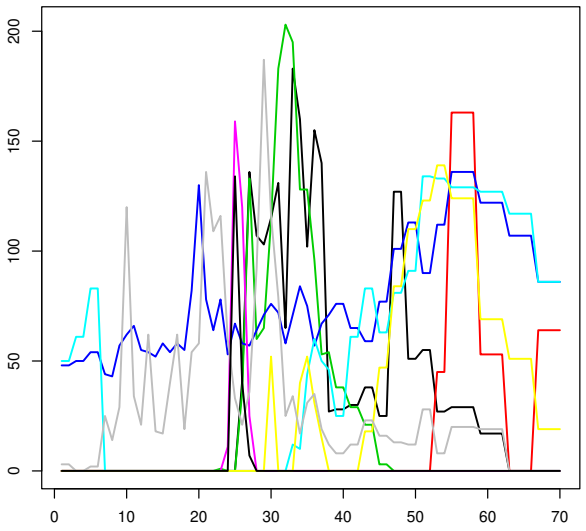


Figure 30: Some curves from a dataset of 472 curves. Each curve is a radar waveform from the Topex/Poseidon satellite.

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.*

[3]See 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`.

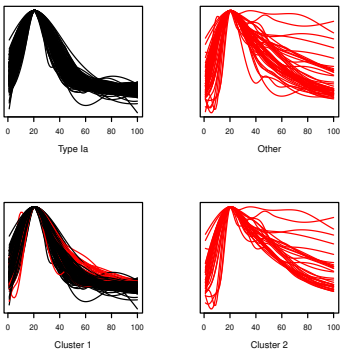


Figure 31: Light curves for supernovae. The top two plots show the light curves for two types of supernovae. The bottom two plots show the results of clustering the curves into two groups, without using knowledge of their labels.

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 [**?**].