

Lecture 1: Course information, supervised vs. unsupervised learning, bias-variance tradeoff

Reading: Chapter 2

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
Jan 19, 2018

Course Information

- ▶ Please read the syllabus carefully.
- ▶ All resources from class will be posted on Canvas
<https://courseworks.columbia.edu/welcome/>. Check the web site often for any important course-related announcements.
- ▶ Lecture meets once every week, from 2:40pm to 5:25pm on Friday.
- ▶ Separate labs/review sessions during the week. Students are REQUIRED to attend ONE lab/review session every week.
- ▶ In each section, students with last name initial from A to L are assigned to the FIRST lecture in the week, and those with last name initial from M to Z are assigned to the SECOND one.

Course Mailing List

We have a course mailing list:

[gr5241_gu4241_course_staff \[at\] columbia \[dot\] edu](mailto:gr5241_gu4241_course_staff@columbia.edu)

For any course-related inquiries, please send them to the mailing list.

Please DO NOT email the instructor or the TAs in person. Any email directly sent to the instructor or the TAs WILL NOT get replied.

Homework

- ▶ We will mainly use R and Python for data analysis.
- ▶ **Homeworks:**
 - a. There will be five assignments. See course schedule for detailed information.
 - b. We **DO NOT** accept late homework.
 - c. The lowest score will be dropped.

Textbook

Textbook

T. Hastie, R. Tibshirani and J. Friedman. *The Elements of Statistical Learning*. Second Edition, Springer, 2009

References

G. James, D. Witten, T. Hastie and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013

K. P. Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

J. Shawe-Taylor and N. Cristianini. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

Grading

Your overall course grade will be determined as a weighted average of the following categories:

5%	attendance
35 %	homework assignments
25 %	midterm exam
35 %	final exam

Exams

- ▶ Midterm

Fri, Mar. 9, 2018, 4:10pm - 5:25pm, *in lecture*

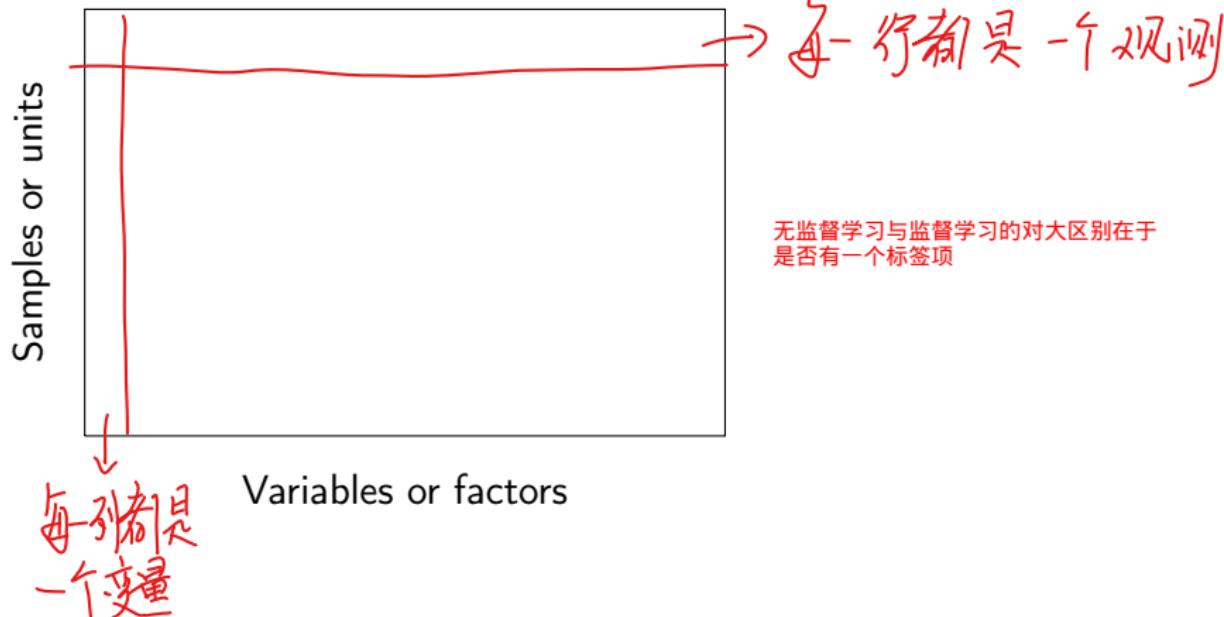
- ▶ Final

Fri, May 4, 2018, 1:10pm - 4:00pm, *in lecture*

- ▶ In general, **NO MAKE-UP EXAMES** are granted. If an emergency occurs on the exam day, you must contact the instructor before the exam.

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:



Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

Samples or units



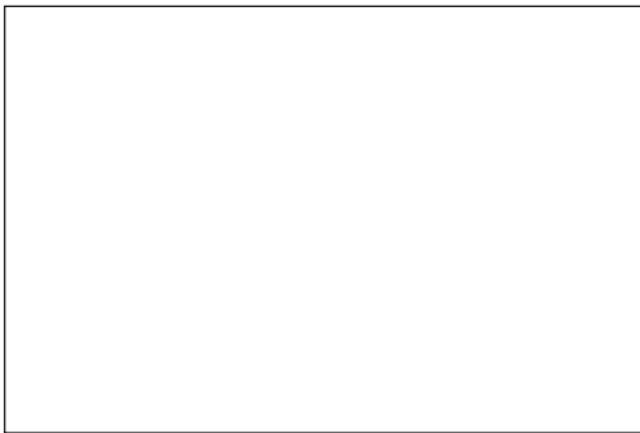
Variables or factors

Quantitative, eg. weight, height, number of children, ...

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

Samples or units



Variables or factors

Qualitative, eg. college major, profession, gender, ...

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

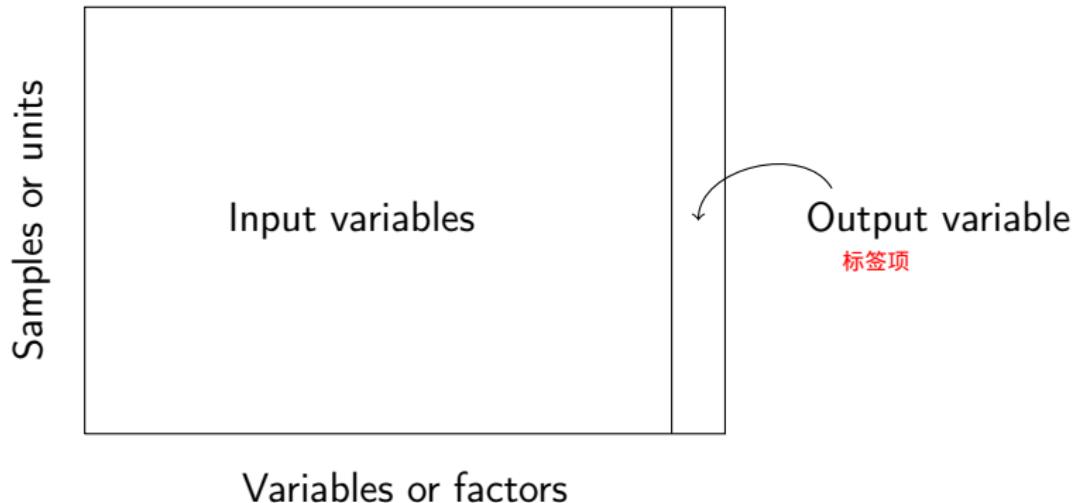
Our goal is to:

- ▶ Find meaningful relationships between the variables or units.
Correlation analysis. *相关分析*
- ▶ Find low-dimensional representations of the data which make it easy to visualize the variables and units. **PCA, ICA, multidimensional scaling, locally linear embeddings, etc.** *降维*
- ▶ Find meaningful groupings of the data. **Clustering.** *聚类*

Unsupervised learning is also known in Statistics as **exploratory data analysis**.

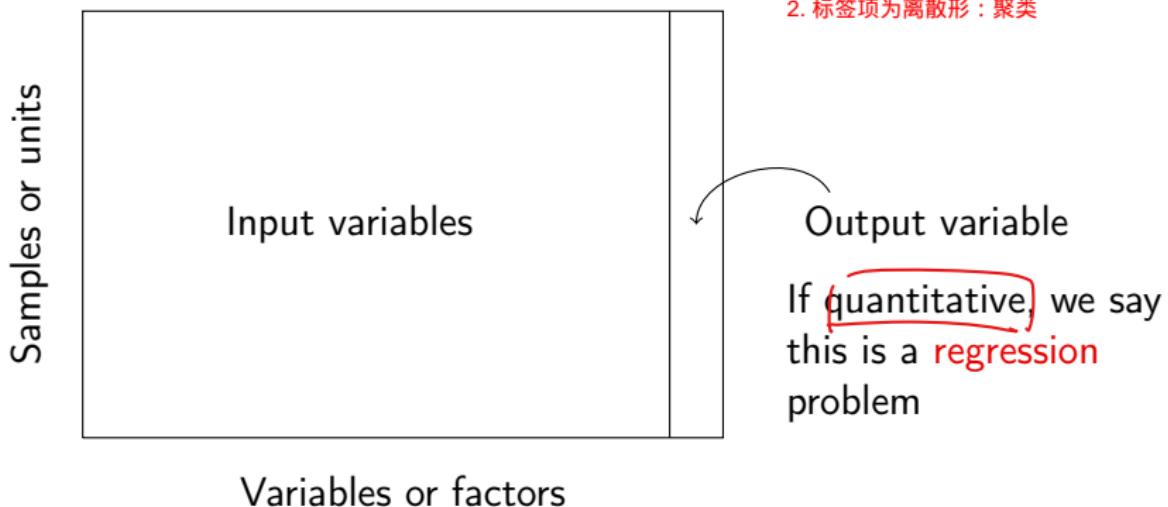
Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:



Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:

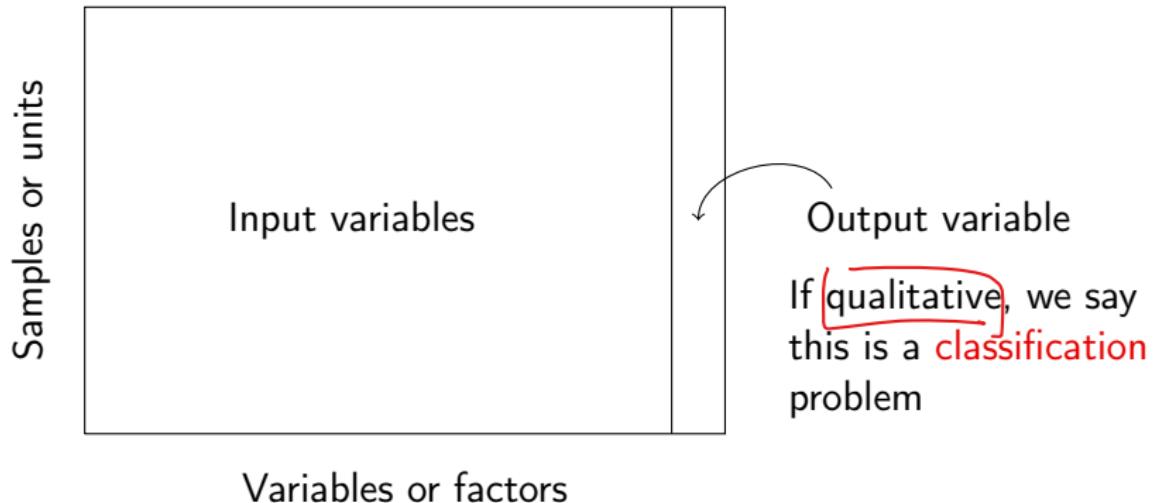


监督学习分为两个主要的类别:

1. 标签项为连续形 : 回归
2. 标签项为离散形 : 聚类

Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:



Supervised vs. unsupervised learning

最大特征

In supervised learning, there are input variables, and output variables:

If X is the vector of inputs for a particular sample. The output variable is modeled by:

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Our goal is to learn the function f , using a set of training samples.

{ training set: estimate the function f
testing set: value the performance of f

Supervised vs. unsupervised learning

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Motivations:

- ▶ **Prediction:** Useful when the input variable is readily available, but the output variable is not.

Example: Predict stock prices next week using data from last month.

Supervised vs. unsupervised learning

$$Y = f(X) + \varepsilon$$

Random error

Motivations:

→ 目的在于估计 研究
X值下的 Y 值

- ▶ **Prediction:** Useful when the input variable is readily available, but the output variable is not.
- ▶ **Inference:** A model for f can help us understand the structure of the data — which variables influence the output, and which don't? What is the relationship between each variable and the output, e.g. linear, non-linear?

Example: What is the influence of genetic variations on the incidence of heart disease.

Parametric and nonparametric methods:

There are two kinds of supervised learning method:

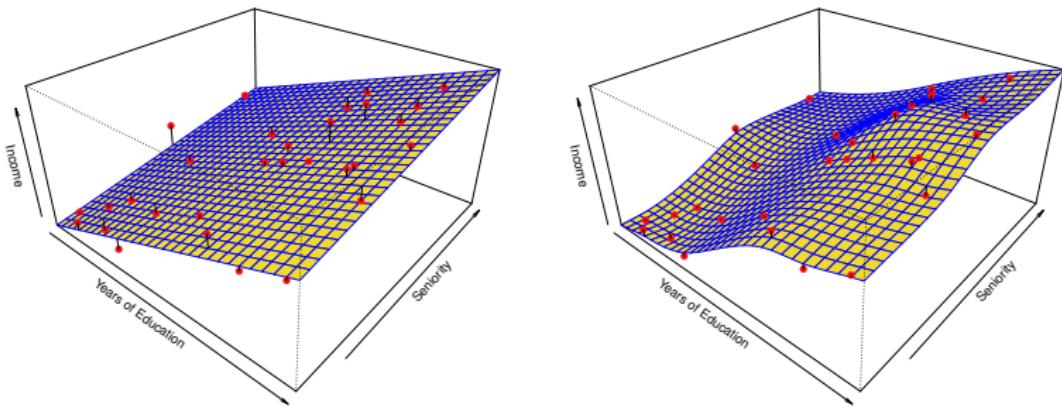
- ▶ **Parametric methods:** We assume that f takes a specific form. For example, a linear form:

$$f(X) = X_1\beta_1 + \cdots + X_p\beta_p$$

with parameters β_1, \dots, β_p . Using the training data, we try to *fit* the parameters.

- ▶ **Non-parametric methods:** We don't make any assumptions on the form of f , but we restrict how "wiggly" or "rough" the function can be.

Parametric vs. nonparametric prediction



ISL Figures 2.4 and 2.5

↑ 易于解决 inference 问题, 但 predict 的精度低

(Parametric methods have a limit of fit quality. Non-parametric methods keep improving as we add more data to fit. ↓)

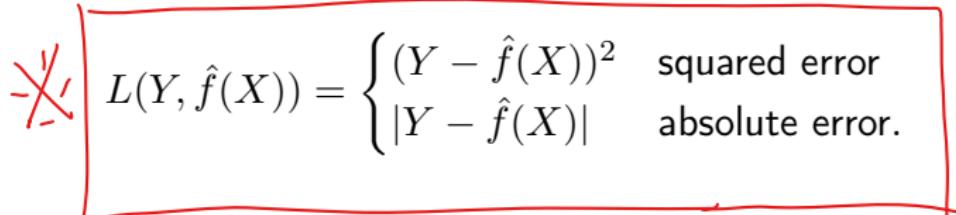
Parametric methods are often simpler to interpret.

易于不断提升 predict
但形式复杂难以 inference

Loss Function

The **loss function** $L(Y, \hat{f}(X))$ measures the errors between the observed value Y and the predicted value $\hat{f}(X)$.

In a regression problem, two most common loss functions are:


$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{squared error} \\ |Y - \hat{f}(X)| & \text{absolute error.} \end{cases}$$

Prediction error

Training data: $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$

Predicted function: \hat{f} .

Our goal in supervised learning is to minimize the expected prediction error. Under squared-error loss, this is the *Mean Squared Error*:

$$MSE(\hat{f}) = E(y_0 - \hat{f}(x_0))^2. = \mathbb{E}[(y_0^2 + \hat{f}^2(x_0) - 2y_0\hat{f}(x_0))]$$

通常無法
的

Unfortunately, this quantity cannot be computed, because we don't know the joint distribution of (X, Y) . We can compute a sample average using the **training data**; this is known as the training MSE:

$$MSE_{\text{training}}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2.$$

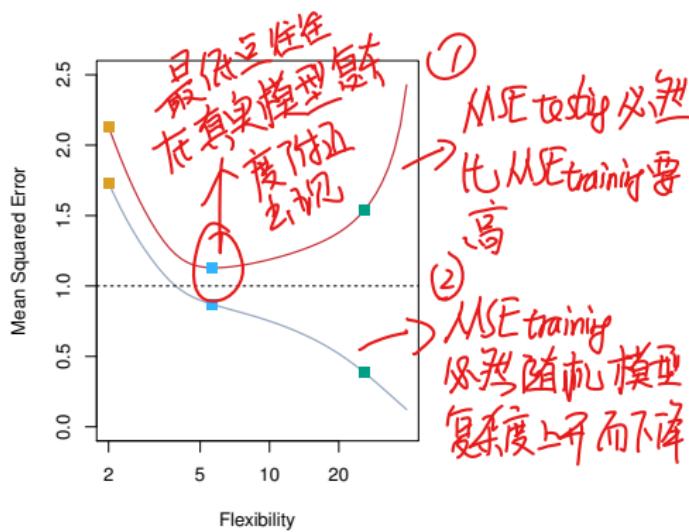
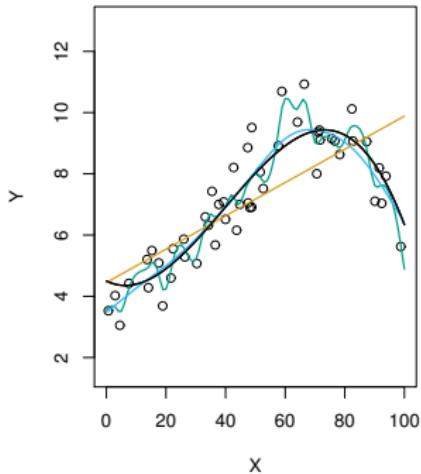
Prediction error

The main challenge of statistical learning is that *a low training MSE does not imply a low MSE.*

If we have test data $\{(x'_i, y'_i); i = 1, \dots, m\}$ which were not used to fit the model, a better measure of quality for \hat{f} is the test MSE:

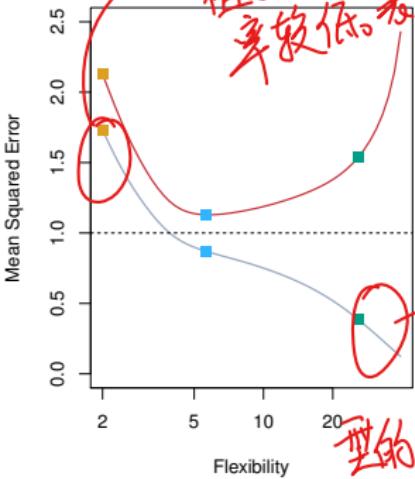
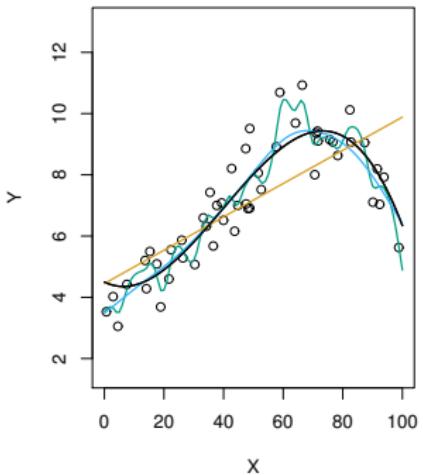
$$MSE_{\text{test}}(\hat{f}) = \frac{1}{m} \sum_{i=1}^m (y'_i - \hat{f}(x'_i))^2.$$

ISL Figure 2.9.



The circles are simulated data from the black curve.

ISL Figure 2.9.

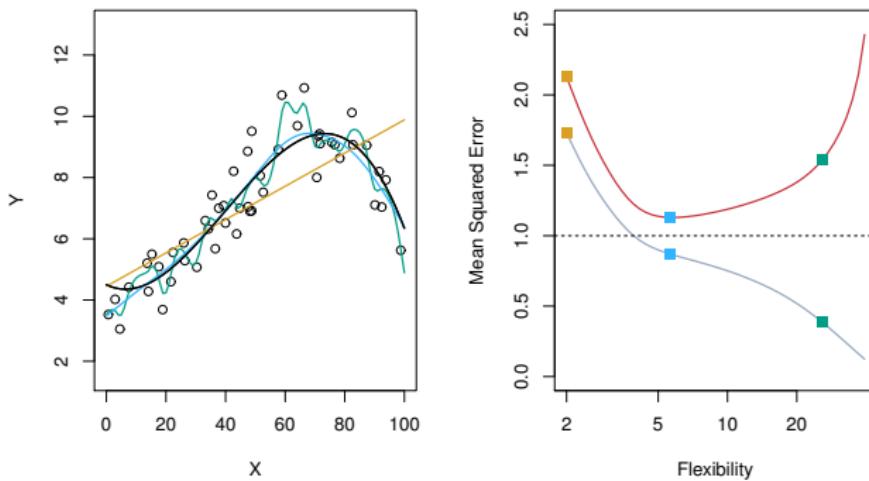


The circles are simulated data from the black curve.
In this artificial example, we know what f is.

→ MSE training
值较大表明模型的准确率较低，称为 high bias

→ MSE training 较小 → 表明对这一组数据模型的准确度较高。
→ 易受到数据影响，预测结果不稳定
→ high variance

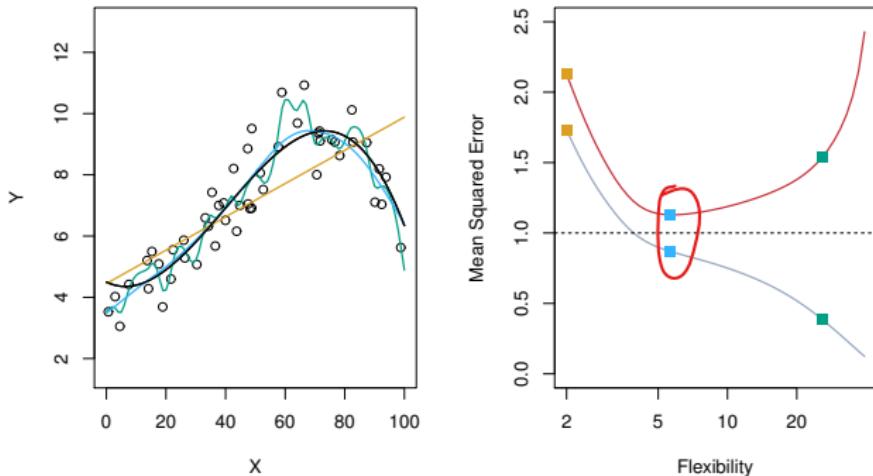
ISL Figure 2.9.



Three estimates \hat{f} are shown:

1. Linear regression.
2. Splines (very smooth).
3. Splines (quite rough).

ISL Figure 2.9.

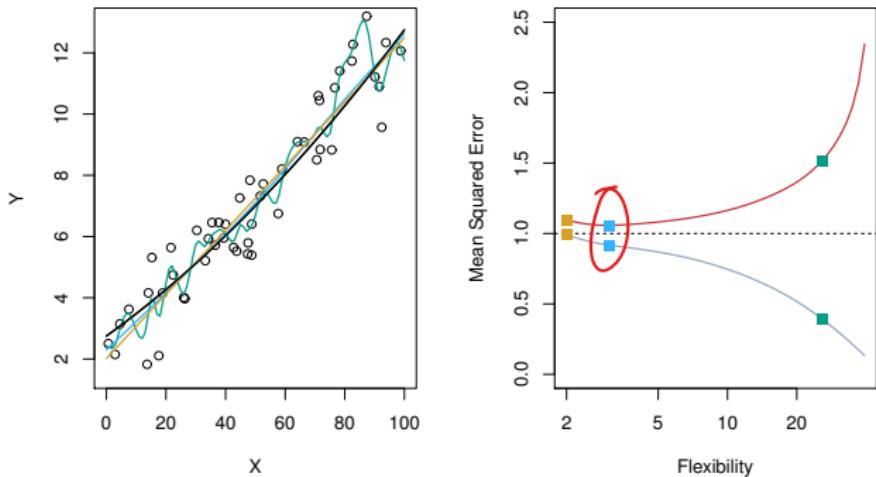


当真实模型中的数据更丰富时
为中时

Red line: Test MSE.

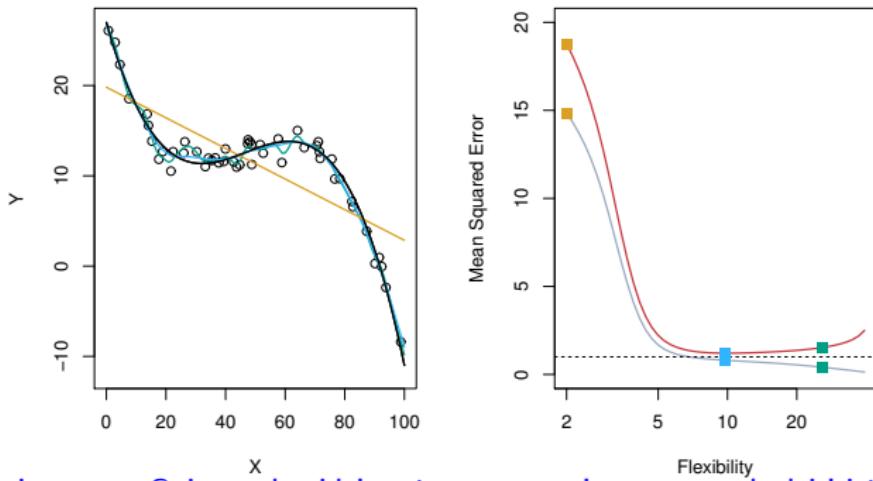
Gray line: Training MSE.

ISL Figure 2.10



The function f is now almost linear.

ISL Figure 2.11



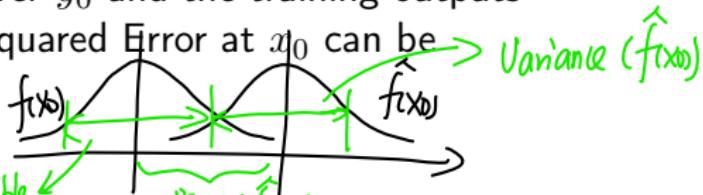
low variance of irreducible term ==> lower probability to be overfit ==> low variance of the model

When the noise ε has small variance, the third method does well.

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and \hat{f} be estimated from n training samples $(x_1, y_1) \dots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs (y_1, \dots, y_n) . Then, the Mean Squared Error at x_0 can be decomposed:



$$\begin{aligned} MSE(x_0) &= E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0). \\ &= E[(y_0 - f(x_0)) + (f(x_0) - \hat{f}(x_0)) + (\hat{f}(x_0) - \hat{f}(x_0))]^2 \\ &= E[\underbrace{[y_0 - f(x_0)]^2}_{\text{irreducible error}} + \underbrace{[f(x_0) - \hat{f}(x_0)]^2}_{\text{Bias}(\hat{f}(x_0))} + \underbrace{\{E(\hat{f}(x_0)) - \hat{f}(x_0)\}^2}_{\text{variance}} + \text{交叉项}] \\ &= E[y_0 - f(x_0)]^2 + E[f(x_0) - \hat{f}(x_0)]^2 + E[E(\hat{f}(x_0)) - \hat{f}(x_0)]^2 + \text{交叉项} \\ &= \text{Var}(\varepsilon_0) \quad \text{[irreducible error]} \quad \text{Bias}(\hat{f}(x_0))^2 \quad E[E(\hat{f}(x_0)) - \hat{f}(x_0)]^2 \\ &\quad \text{variance}(\hat{f}(x_0)) \end{aligned}$$

交叉验证 0:

The bias variance decomposition

$$\begin{aligned} & \underbrace{\mathbb{E}(y_0 - f(x_0))}_{\mathbb{E}(y_0) - f(x_0)} \underbrace{\mathbb{E}(f(x_0) - \mathbb{E}(\hat{f}(x_0)))}_{\mathbb{E}(\hat{f}(x_0)) - f(x_0)} \\ &= \mathbb{E}(y_0) - f(x_0) = 0 \end{aligned}$$

$$\begin{aligned} & \mathbb{E}[\mathbb{E}(\hat{f}(x_0)) - f(x_0)] \\ & \times \mathbb{E}(y_0 - \hat{f}(x_0)) \\ &= 0 \end{aligned}$$

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and \hat{f} be estimated from n training samples $(x_1, y_1) \dots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs (y_1, \dots, y_n) . Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

Irreducible error

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and \hat{f} be estimated from n training samples $(x_1, y_1) \dots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs (y_1, \dots, y_n) . Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

The variance of the estimate of Y : $E[\hat{f}(x_0) - E(\hat{f}(x_0))]^2$

This measures how much the estimate of \hat{f} at x_0 changes when we sample new training data.

预测的稳定性

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and \hat{f} be estimated from n training samples $(x_1, y_1) \dots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs (y_1, \dots, y_n) . Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

The squared bias of the estimate of Y : $[E(\hat{f}(x_0)) - f(x_0)]^2$

This measures the deviation of the average prediction $\hat{f}(x_0)$ from the truth $f(x_0)$.

预测的平均准确性

Implications of bias variance decomposition

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon).$$

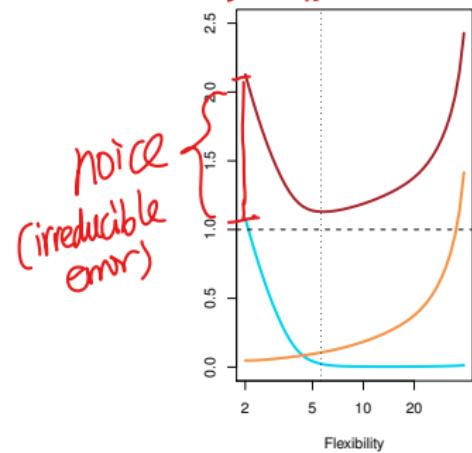
- ▶ The MSE is always positive.
- ▶ Each element on the right hand side is always positive.
- ▶ Therefore, typically when we decrease the bias beyond some point, we increase the variance, and vice-versa.

More flexibility \iff Higher variance \iff Lower bias.



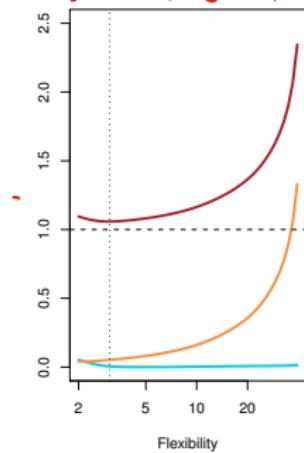
Squiggly f , high noise

根据本模型规律较复杂
且变动较大



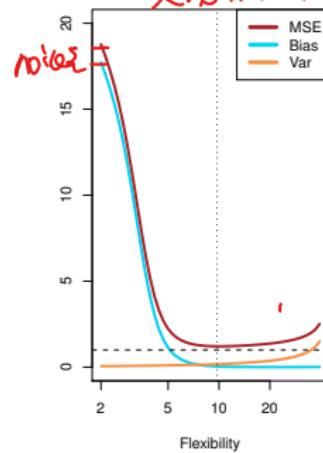
Linear f , high noise

根据本模型规律较简单
且变动较大



Squiggly f , low noise

根据本模型规律较复杂
但变动不大



ISL Figure 2.12

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set {Ford, Toyota, Mercedes-Benz, ...}.

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set $\{\text{Ford, Toyota, Mercedes-Benz, ...}\}$.

The model:

$$Y = f(X) + \varepsilon$$

becomes insufficient, as f is not necessarily real-valued.

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set {Ford, Toyota, Mercedes-Benz, ...}.

The model:

$$\underline{Y = f(X) + \varepsilon}$$

becomes insufficient, as f is not necessarily real-valued.

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set $\{\text{Ford, Toyota, Mercedes-Benz, ...}\}$.

We will use slightly different notation:

$P(X, Y)$: joint distribution of (X, Y) ,

$P(Y | X)$: conditional distribution of X given Y ,

\hat{y}_i : prediction for x_i .

Loss function for classification

There are many ways to measure the error of a classification prediction. One of the most common is the 0-1 loss:

$$L_0-1 = E(\mathbf{1}(y_0 \neq \hat{y}_0))$$

Like the MSE, this quantity can be estimated from training and test data by taking a sample average:

$$\left[\frac{1}{n} \sum_{i=1}^n \mathbf{1}(y_i \neq \hat{y}_i) \right] \rightarrow \text{错误率}$$

Bayes classifier

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009. Chap 2

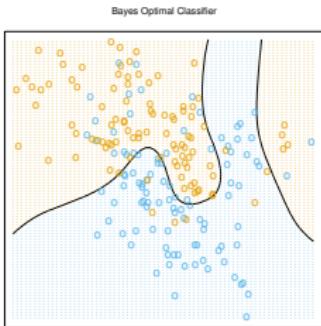


FIGURE 2.5. The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).

In practice, we never know the joint probability P . However, we can assume that it exists.

The **Bayes classifier** assigns:

$$\hat{y}_i = \operatorname{argmax}_k P(Y = k \mid X = x_i)$$

It can be shown that this is the best classifier under the 0-1 loss.

proof that under 0-1 Loss, bayes classifier is optimal:

$$R(f|x) = \bar{\sum} E(\mathbb{I}(y_0 \neq \hat{y}_0) P(y|x)).$$

$$= \sum_{y \neq f(x)} 1xP(y|x) + \sum_{y=f(x)} 0xP(y|x)$$

$$= \sum_{y \neq f(x)} P(y|x) = 1 - P(f(x)|x)$$

$$\hat{f}_0 = \arg \max_{[K]} P(y=k|x=x)$$

(贝叶斯决策)

$$\Rightarrow \hat{f}_0 \text{ minimize } R(f|x).$$

Thanks to Sergio Bacallado and Peter Orbanz

给定 x ,
 $P(y)$ 在 $y=f(x)$ 下的
概率加上 $P(y)$ 在 $y \neq f(x)$
下的概率等于 1

for sharing the slides.

$$\Rightarrow R(f) = \int R(f(x)) P(x) dx$$

$$\Rightarrow \hat{f}_0 \text{ minimize } R(f)$$

where $R(f)$ is the
expectation of 0-1 Loss.

Lecture 2: Frequency Vs. Bayes

Reading: Sections 2.4, 8.3

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
January 19, 2018

Parametric Models

Models

A **model** \mathcal{P} is a set of probability distributions. We index each distribution by a parameter value $\theta \in \mathcal{T}$; we can then write the model as

$$\mathcal{P} = \{P_\theta | \theta \in \mathcal{T}\}.$$

The set \mathcal{T} is called the **parameter space** of the model.

Parametric model

The model is called **parametric** if the number of parameters (i.e. the dimension of the vector θ) is (1) finite and (2) independent of the number of data points. Intuitively, the complexity of a parametric model does not increase with sample size.

Density representation

For parametric models, we can assume that $\mathcal{T} \subset \mathbb{R}^d$ for some fixed dimension d . We usually represent each P_θ be a density function $p(x|\theta)$.

Maximum Likelihood Estimation

Setting

- ▶ Given: Data x_1, \dots, x_n , parametric model $\mathcal{P} = \{p(x|\theta) \mid \theta \in \mathcal{T}\}$.
- ▶ Objective: Find the distribution in \mathcal{P} which best explains the data.
That means we have to choose a "best" parameter value $\hat{\theta}$.

Maximum Likelihood approach

Maximum Likelihood assumes that the data is best explained by the distribution in \mathcal{P} under which it has the highest probability (or highest density value).

Hence, the **maximum likelihood estimator** is defined as

$$\hat{\theta}_{ML} := \arg \max_{\theta \in \mathcal{T}} p(x_1, \dots, x_n | \theta)$$

the parameter which maximizes the joint density of the data.

Analytic Maximum Likelihood

The i.i.d. assumption

The standard assumption of ML methods is that the data is **independent and identically distributed (i.i.d.)**, that is, generated by independently sampling repeatedly from the same distribution P .

If the density of P is $p(x|\theta)$, that means the joint density decomposes as

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i|\theta)$$


Maximum Likelihood equation

The analytic criterion for a maximum likelihood estimator (under the i.i.d. assumption) is:

$$\nabla_{\theta} \left(\prod_{i=1}^n p(x_i|\theta) \right) = 0$$


We use the "logarithm trick" to avoid a huge product rule computation.

Logarithm Trick

Recall: Logarithms turn products into sums

$$\log\left(\prod_i f_i\right) = \sum_i \log(f_i)$$

Logarithms and maxima

The logarithm is monotonically increasing on \mathbb{R}_+ .

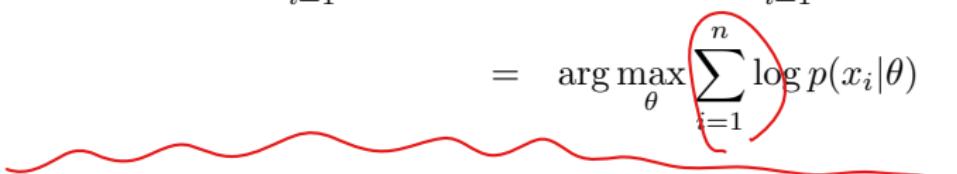
Consequence: Application of log does not change the *location* of a maximum or minimum:

$$\max_y \log(g(y)) \neq \max_y g(y) \quad \text{The } \textit{value} \text{ changes.}$$

$$\arg \max_y \log(g(y)) = \arg \max_y g(y) \quad \text{The } \textit{location} \text{ does not change.}$$

Analytic MLE

Likelihood and logarithm trick

$$\begin{aligned}\hat{\theta}_{ML} &= \arg \max_{\theta} \prod_{i=1}^n p(x_i | \theta) = \arg \max_{\theta} \log \left(\prod_{i=1}^n p(x_i | \theta) \right) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log p(x_i | \theta)\end{aligned}$$


Analytic maximality criterion

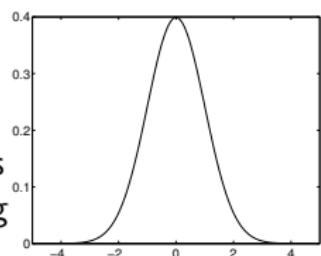
$$0 = \sum_{i=1}^n \nabla_{\theta} \log p(x_i | \theta) = \sum_{i=1}^n \frac{\nabla_{\theta} p(x_i | \theta)}{p(x_i | \theta)}$$

Whether or not we can solve this analytically depends on the choice of the model!

Example: Gaussian Mean MLE

Gaussian density in one dimension

$$g(x; \mu, \sigma) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$



- The quotient $\frac{x-\mu}{\sigma}$ measures deviation of x from its expected value in units of σ (i.e. σ defines the length scale)

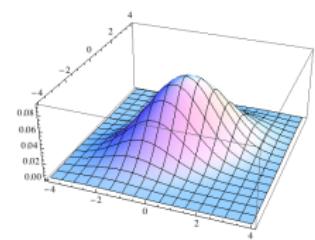
Gaussian density in d dimensions

The quadratic function

$$-\frac{(x - \mu)^2}{2\sigma^2} = -\frac{1}{2}(x - \mu)(\sigma^2)^{-1}(x - \mu)$$

is replaced by a quadratic form:

$$g(\mathbf{x}; \boldsymbol{\mu}, \Sigma) := \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2} \langle (\mathbf{x} - \boldsymbol{\mu}), \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) \rangle\right)$$



Example: Gaussian Mean MLE

Model: Multivariate Gaussians

The model \mathcal{P} is the set of all Gaussian densities on \mathbb{R}^d with *fixed* covariance matrix Σ ,

$$\mathcal{P} = \{g(\cdot | \mu, \Sigma) \mid \mu \in \mathbb{R}^d\},$$

where g is the Gaussian density function. The parameter space is $\mathcal{T} = \mathbb{R}^d$.

MLE equation

We have to solve the maximum equation

$$\sum_{i=1}^n \nabla_\mu \log g(x_i | \mu, \Sigma) = 0$$

for μ .

Example: Gaussian Mean MLE

$$\begin{aligned} 0 &= \sum_{i=1}^n \nabla_\mu \log \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2} \langle (x_i - \mu), \Sigma^{-1}(x_i - \mu) \rangle\right) \\ &= \sum_{i=1}^n \nabla_\mu \left(\log\left(\frac{1}{\sqrt{(2\pi)^d |\Sigma|}}\right) + \log\left(\exp\left(-\frac{1}{2} \langle (x_i - \mu), \Sigma^{-1}(x_i - \mu) \rangle\right)\right) \right. \\ &\quad \left. \text{与 } \mu \text{ 无关, 求导时直接忽略} \right) \\ &= \sum_{i=1}^n \nabla_\mu \left(-\frac{1}{2} \langle (x_i - \mu), \Sigma^{-1}(x_i - \mu) \rangle \right) = -\sum_{i=1}^n \Sigma^{-1}(x_i - \mu) \end{aligned}$$

Multiplication by $(-\Sigma)$ gives

$$0 = \sum_{i=1}^n (x_i - \mu) \quad \Rightarrow \quad \mu = \frac{1}{n} \sum_{i=1}^n x_i$$

Conclusion

The maximum likelihood estimator of the Gaussian expectation parameter for fixed covariance is

$$\hat{\mu}_{\text{ML}} := \frac{1}{n} \sum_{i=1}^n x_i$$

Example: Gaussian with Unknown Covariance

Model: Multivariate Gaussians

The model \mathcal{P} is now

$$\mathcal{P} = \{g(\cdot | \mu, \Sigma) \mid \mu \in \mathbb{R}^d, \Sigma \in \Delta_d\},$$

where Δ_d is the set of positive definite $d \times d$ -matrices. The parameter space is $\mathcal{T} = \mathbb{R}^d \times \Delta_d$.

ML approach

Since we have just seen that the ML estimator of μ does not depend on Σ , we can compute $\hat{\mu}_{\text{ML}}$ first. We then estimate Σ using the criterion

$$\sum_{i=1}^n \nabla_{\Sigma} \log g(x_i | \hat{\mu}_{\text{ML}}, \Sigma) = 0$$

Solution

The ML estimator of Σ is

$$\hat{\Sigma}_{\text{ML}} := \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_{\text{ML}})(x_i - \hat{\mu}_{\text{ML}})^t.$$

Bayesian models

The defining assumption of **Bayesian statistics** is that the distribution P_θ which models the data is a **random quantity** and itself has a distribution Q . The generative model for data X_1, X_2, \dots is

$$\begin{aligned} P_\theta &\sim Q \\ X_1, X_2, \dots &\stackrel{i.i.d.}{\sim} P_\theta \end{aligned}$$

The rational behind the approach is:

- ▶ In any statistical approach (Bayesian or frequentist), the distribution P_θ is unknown.
- ▶ Bayesian statistics argues that any form of uncertainty should be expressed by probability distributions.
- ▶ We can think of the randomness in Q as a model of the statistician's lack of knowledge regarding P_θ .

Prior and posterior

The distribution Q of P_θ is called the **a priori distribution** (or the **prior** for short). We use q to denote its density if it exists.

Our objective is to determine the conditional probability of P given observed data

$$\Pr(\theta|x_1, \dots, x_n).$$

The distribution is called the **a posteriori distribution** or **posterior**.

Bayes' Theorem

Given data X_1, \dots, X_n , we can compute the posterior by

$$\Pr(\theta|x_1, \dots, x_n) = \frac{(\prod_{i=1}^n p(x_i|\theta))q(\theta)}{p(x_1, \dots, x_n)} = \frac{(\prod_{i=1}^n p(x_i|\theta))q(\theta)}{\int (\prod_{i=1}^n p(x_i|\theta)) q(\theta)}.$$

The individual terms have names:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Example: unknown Gaussian mean

Model

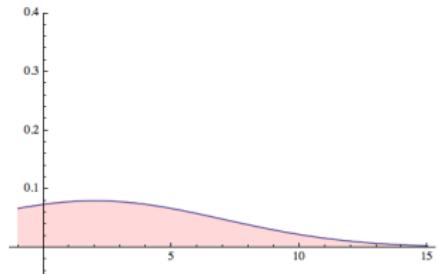
We assume that the data is generated from a Gaussian with fixed variance σ^2 . The mean μ is unknown. The model likelihood is $p(x|\mu, \sigma) = g(x|\mu, \sigma)$ (where g is the Gaussian density on the line).

Bayesian model

We choose a Gaussian prior on μ ,

$$q(\mu) := g(\mu|\mu_0, \sigma_0) .$$

In the figure, $\mu_0 = 2$ and $\sigma_0 = 5$. Hence, we assume that $\mu_0 = 2$ is the most probable value of μ , and that $\mu \in [-3, 7]$ with a probability ~ 0.68 .



Example: unknown Gaussian mean

Application of Bayes' formula to the Gaussian-Gaussian model shows the posterior distribution is

$$\Pr(\mu|x_{1:n}) = g(\mu|\mu_n, \sigma_n),$$

where $\mu_n := \frac{\sigma^2 \mu_0 + \sigma_0^2 \sum_{i=1}^n x_i}{\sigma^2 + n\sigma_0^2}$ and $\sigma_n^2 := \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}$. \rightarrow 随着 $n \uparrow \Rightarrow$ 越来越小.

$$\mu_n = \frac{\sigma^2}{\sigma^2 + n\sigma_0^2} \mu_0 + \frac{n\sigma_0^2}{\sigma^2 + n\sigma_0^2} \bar{x}_i \quad (\text{weighted average}).$$

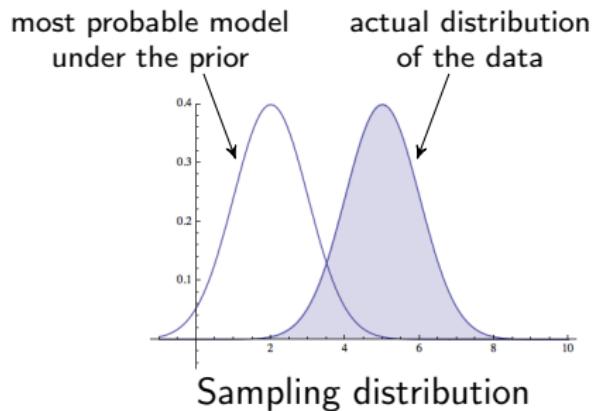
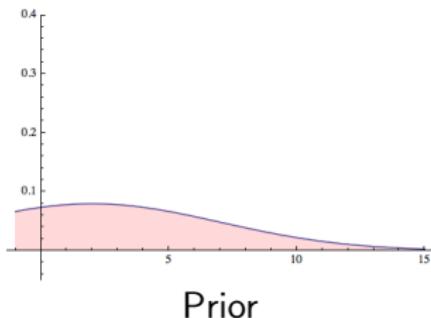
\rightarrow $n \uparrow \Rightarrow$ \bar{x}_i 的权重↑

Example: unknown Gaussian mean

Application of Bayes' formula to the Gaussian-Gaussian model shows the posterior distribution is

$$\Pr(\mu|x_{1:n}) = g(\mu|\mu_n, \sigma_n),$$

$$\text{where } \mu_n := \frac{\sigma^2 \mu_0 + \sigma_0^2 \sum_{i=1}^n x_i}{\sigma^2 + n\sigma_0^2} \text{ and } \sigma_n^2 := \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}.$$

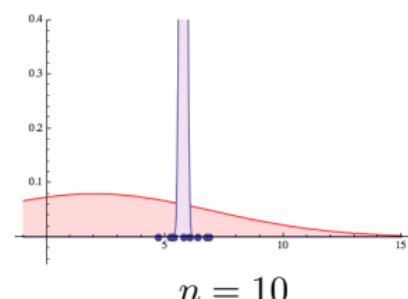
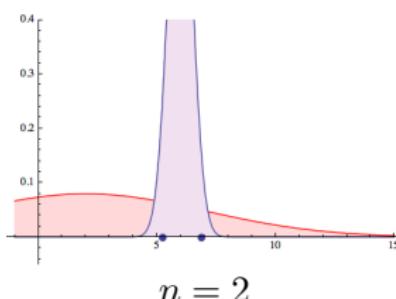
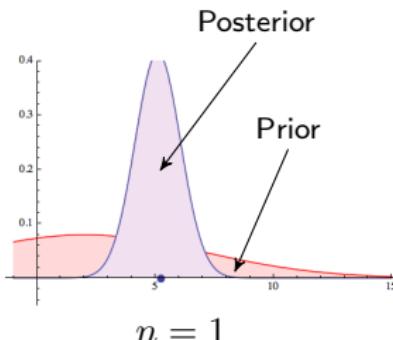


Example: unknown Gaussian mean

Application of Bayes' formula to the Gaussian-Gaussian model shows the posterior distribution is

$$\Pr(\mu|x_{1:n}) = g(\mu|\mu_n, \sigma_n),$$

$$\text{where } \mu_n := \frac{\sigma^2 \mu_0 + \sigma_0^2 \sum_{i=1}^n x_i}{\sigma^2 + n\sigma_0^2} \text{ and } \sigma_n^2 := \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}.$$



MAP estimation

Suppose $\Pi(\theta|x_{1:n})$ is the posterior of a Bayesian model. The estimator

$$\hat{\theta}_{\text{MAP}} = \arg \max_{\theta} \Pi(\theta|x_{1:n})$$

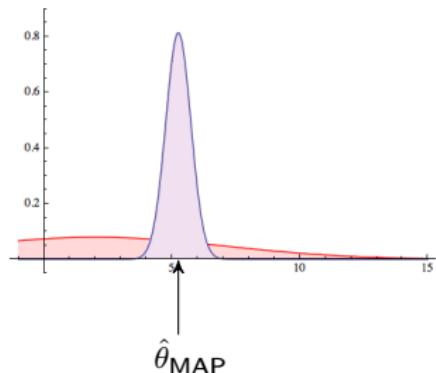
本质上就是 posterior 的 MLE

is called the **maximum a posteriori** (or **MAP**) estimator for θ .

Point estimates

The goal of Bayesian inference is to compute the posterior distribution. Contrast this to classical statistics (e.g. maximum likelihood), where we typically estimate a single value for θ (a so-called **point estimate**).

MAP estimation combines aspects of Bayesian methodology (use of a prior) with aspects of classical methodology (since $\hat{\theta}_{\text{MAP}}$ is a point estimate).



MAP and regularization

Logarithmic view

Since the logarithm leaves the maximum invariant,

$$\hat{\theta}_{\text{MAP}} = \arg \max_{\theta} \Pi(\theta | x_{1:n}) = \arg \max_{\theta} \log \Pi(\theta | x_{1:n})$$

Substituting in the Bayes equation gives $\Pi(\theta | x_1, \dots, x_n) = \frac{p(\theta) \times p(x_1, \dots, x_n | \theta)}{\int p(\theta) \times p(x_1, \dots, x_n | \theta) d\theta}$

$$\log \Pi(\theta | x_{1:n}) = \sum_{i=1}^n \log p(x_i | \theta) + \log q(\theta) - \log p(x_1, \dots, x_n)$$

MAP as regularized ML

Since log-evidence does not depend on θ ,

$$\hat{\theta}_{\text{MAP}} = \arg \max_{\theta} \left\{ \sum_{i=1}^n \log p(x_i | \theta) + \log q(\theta) \right\}$$

Thus, the MAP estimate can be regarded as a regularized version of a maximum likelihood estimator. The regularization term $\log q(\theta)$ favors values where q (and hence $\log q$) is large.

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set
 $\{\text{Ford, Toyota, Mercedes-Benz, ...}\}$.

We will use slightly different notation:

$P(X, Y)$: joint distribution of (X, Y) ,

$P(Y | X)$: conditional distribution of X given Y ,

\hat{y}_i : prediction for x_i .

Loss function for classification

There are many ways to measure the error of a classification prediction.
One of the most common is the 0-1 loss:

$$E(\mathbf{1}(y_0 \neq \hat{y}_0))$$

Like the MSE, this quantity can be estimated from training and test data by taking a sample average:

$$\frac{1}{n} \sum_{i=1}^n \mathbf{1}(y_i \neq \hat{y}_i)$$

Bayes classifier

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009. Chap 2

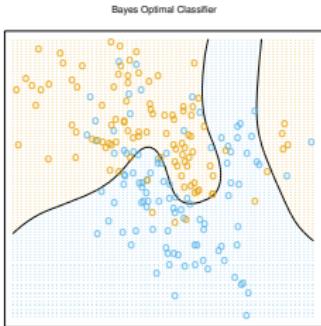


FIGURE 2.5. The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).

In practice, we never know the joint probability P . However, we can assume that it exists.

The **Bayes classifier** assigns:

$$\hat{y}_i = \operatorname{argmax}_k P(Y = k \mid X = x_i)$$

It can be shown that this is the best classifier under the 0-1 loss.

Images of Linear Mappings (1)

$|X^n$
 $y = X(A)$
 $|X^m \times |n|$
通过A将X
映射到Y.

Linear mapping

A matrix $X \in \mathbb{R}^{n \times m}$ defines a linear mapping $f_x : \mathbb{R}^m \rightarrow \mathbb{R}^n$.

Image

Recall: The **image** of a mapping f is the set of all possible function values, here

投影之后Y的可能取值

$$\text{image}(f_x) := \{y \in \mathbb{R}^n \mid Xz = y \text{ for some } z \in \mathbb{R}^m\}$$

Image of a linear mapping

- ▶ The image of a linear mapping $\mathbb{R}^m \rightarrow \mathbb{R}^n$ is a linear subspace of \mathbb{R}^n .
- ▶ The columns of X form a basis of the image space:

$$\text{image}(\tilde{X}) = \text{span}\{\tilde{X}_1^{\text{col}}, \dots, \tilde{X}_m^{\text{col}}\}$$

- ▶ This is one of most useful things to remember about matrices, so, again:

The columns span the image.

X的范围跨足覆盖
image的范围.

Images of Linear Mappings (2)

Dimension of the image space

Clearly: The number of linearly independent column vectors. This number is called the column rank of $\tilde{\mathbf{X}}$.

Invertible mappings

Recall: A mapping f is invertible if it is one-to-one, i.e. for each function value $\tilde{\mathbf{y}}$ there is exactly one input value with $f(\mathbf{z}) = \tilde{\mathbf{y}}$.

Invertible matrices

The matrix $\tilde{\mathbf{X}}$ is called invertible if $f_{\tilde{\mathbf{X}}}$ is invertible.

- ▶ Only square matrices can be invertible. (反例).
- ▶ For a *linear* mapping: If $\tilde{\mathbf{X}}$ is a square matrix $f_{\tilde{\mathbf{X}}}$ is invertible iff the image has the same dimension as the input space.
- ▶ Even if $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times m}$, the matrix $\tilde{\mathbf{X}}^t \tilde{\mathbf{X}}$ is in $\mathbb{R}^{m \times m}$ (a square matrix).
- ▶ So: $\tilde{\mathbf{X}}^t \tilde{\mathbf{X}}$ is invertible if $\tilde{\mathbf{X}}$ has full column rank.

Symmetric and Orthogonal Matrices

Recall: Transpose

The transpose A^T of a matrix $A \in \mathbb{R}^{m \times m}$ is the matrix with entries

$$(A^T)_{ij} := A_{ji}$$

Orthogonal matrices

A matrix $O \in \mathbb{R}^{m \times m}$ is called **orthogonal**

$$O^{-1} = O^T$$

Orthogonal matrices describe two types of operations:

1. Rotations of the coordinate system.
2. Permutations of the coordinate axes.

Symmetric matrices

A matrix $A \in \mathbb{R}^{m \times m}$ is called **symmetric**

$$A = A^T$$

Note: Symmetric and orthogonal matrices are very different objects. Only the identity is both.

Orthonormal Bases

Recall: ONB

A basis $\{v_1, \dots, v_m\}$ of \mathbb{R}^m is called an **orthonormal basis** if

$$\langle v_i, v_j \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

In other words, the v_i are pairwise orthogonal and each of length 1.

Orthogonal matrices

A matrix is orthogonal precisely if its rows form an ONB. Any two ONBs can be transformed into each other by an orthogonal matrix.

Basis representation

Representation of a vector

Suppose $\mathcal{E} = \{e_1, \dots, e_d\}$ is a basis of a vector space. Then a vector x is represented as

do: 正基

$$x = \sum_{j=1}^d [x_j]_{\mathcal{E}} e^{(j)}$$

$\times = \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix} \rightarrow \text{该正基下的坐标表示}$

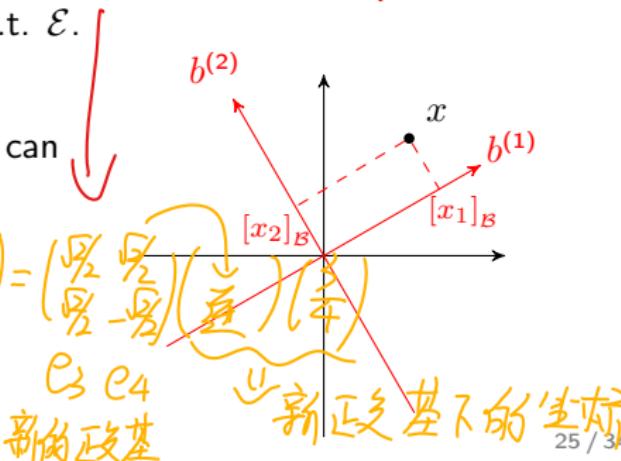
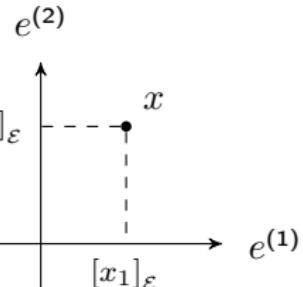
$[x_j]_{\mathcal{E}} \in \mathbb{R}$ are the coordinates of x w.r.t. \mathcal{E} .

Other bases

If $\mathcal{B} = \{b_1, \dots, b_d\}$ is another basis, x can be represented alternatively as

$$x = \sum_{j=1}^d [x_j]_{\mathcal{B}} b^{(j)}$$

$\times' = \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix} \rightarrow \text{新正基下的坐标表示}$



Changing bases

Change-of-basis matrix

The matrix

$$M := \left([e^{(1)}]_{\mathcal{B}}, \dots, [e^{(d)}]_{\mathcal{B}} \right)$$

transforms between the bases, i.e.

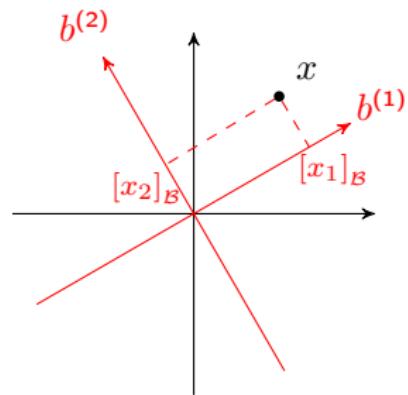
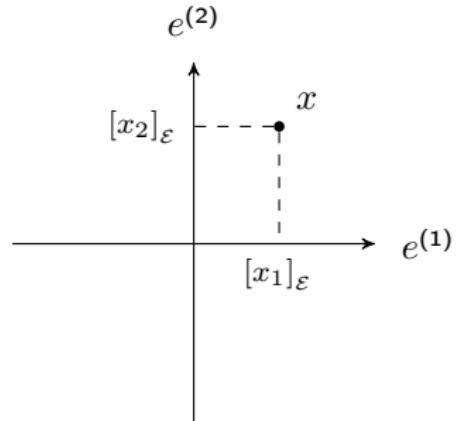
$$M[x]_{\mathcal{E}} = [x]_{\mathcal{B}} .$$

If both \mathcal{E} and \mathcal{B} are ONBs, M is orthogonal.

Representation of matrices

The matrix representing a linear mapping $A : \mathbb{R}^d \rightarrow \mathbb{R}^d$ in the basis \mathcal{E} is computed as

$$[A]_{\mathcal{E}} := \left([A(e^{(1)})]_{\mathcal{E}}, \dots, [A(e^{(d)})]_{\mathcal{E}} \right)$$



Basis Change for Linear Mappings

Transforming matrices

The matrix representing a linear mapping also changes when we change bases:

$$[A]_{\mathcal{B}} = M[A]_{\mathcal{E}} M^{-1}.$$

Applied to a vector x , this means:

$$[A]_{\mathcal{B}}[x]_{\mathcal{B}} = M[A]_{\mathcal{E}}M^{-1}[x]_{\mathcal{B}}.$$

apply A in representation \mathcal{E}

↓

transform x from \mathcal{B} to \mathcal{E}

↑

transform x back to \mathcal{B}

Transforming between ONBs

If $\mathcal{V} = \{v_1, \dots, v_m\}$ and $\mathcal{W} = \{w_1, \dots, w_m\}$ are any two ONBs, there is an orthogonal matrix O such that

$$[A]_{\mathcal{V}} = O[A]_{\mathcal{W}}O^{-1}$$

for any linear mapping A .

Eigenvalues

We consider a square matrix $A \in \mathbb{R}^{m \times m}$.

Definition

A vector $\xi \in \mathbb{R}^m$ is called an **eigenvector** of A if the direction of ξ does not change under application of A . In other words, if there is a scalar λ such that

$$A\xi = \lambda\xi.$$

λ is called an **eigenvalue** of A for the eigenvector ξ .

Properties in general

- ▶ In general, eigenvalues are complex numbers $\lambda \in \mathbb{C}$.
- ▶ The class of matrices with the nicest eigen-structure are symmetric matrices, for which all eigenvectors are mutually orthogonal.

该阵的特征向量之间相互正交

Eigenstructure of symmetric matrices

If a matrix is symmetric:

- ▶ There are $\text{rank}(A)$ distinct eigendirections.
- ▶ The eigenvectors are pair-wise orthogonal.

- ▶ If $\text{rank}(A) = m$, there is an ONB of \mathbb{R}^m consisting of eigenvectors of A .

Definiteness

type	if ...
positive definite	all eigenvalues > 0
positive semi-definite	all eigenvalues ≥ 0
negative semi-definite	all eigenvalues ≤ 0
negative definite	all eigenvalues < 0
indefinite	none of the above

Orthonormal Bases

Recall: ONB

A basis $\{v_1, \dots, v_m\}$ of \mathbb{R}^m is called an **orthonormal basis** if

$$\langle v_i, v_j \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

In other words, the v_i are pairwise orthogonal and each of length 1.

Orthogonal matrices

A matrix is orthogonal precisely if its rows form an ONB. Any two ONBs can be transformed into each other by an orthogonal matrix.

Transforming between ONBs

If $\mathcal{V} = \{v_1, \dots, v_m\}$ and $\mathcal{W} = \{w_1, \dots, w_m\}$ are ONBs, there is an orthogonal matrix O such that

$$A_{[\mathcal{V}]} = O A_{[\mathcal{W}]} O^{-1}$$

for any matrix A . By $A_{[\mathcal{V}]}$, we denote the representation of A in \mathcal{V} .

Eigenvector ONB

Setting

A 是一个对称阵

- Suppose A symmetric, ξ_1, \dots, ξ_m are eigenvectors and form an ONB.
- $\lambda_1, \dots, \lambda_m$ are the corresponding eigenvalues.

How does A act on a vector $v \in \mathbb{R}^m$?

- Represent v in basis ξ_1, \dots, ξ_m :

一个矩阵乘以一个
(正交)对称阵, 实际上是转化为
该矩阵的正基(其特征向量表示). $v = \sum_{j=1}^m v_j^A \xi_j$ $\forall \xi_1, \dots, \xi_m$ 为基的形式来表示.
且 v_j^A 是 A 的特征值
和特征值的次序.

- Multiply by A : Eigenvector definition (recall: $A\xi_j = \lambda_j \xi_j$) yields

$$Av = A\left(\sum_{j=1}^m v_j^A \xi_j\right) = \sum_{j=1}^m v_j^A A\xi_j = \sum_{j=1}^m v_j^A \lambda_j \xi_j$$

Conclusion

$A^2 V = \sum_{j=1}^m v_j^2 \lambda_j^2 \xi_j \Rightarrow A^n V = \sum_{j=1}^m v_j^n \lambda_j^n \xi_j \rightarrow$ 方向不变
A symmetric matrix acts by scaling the directions ξ_j 放大 n 倍

Illustration

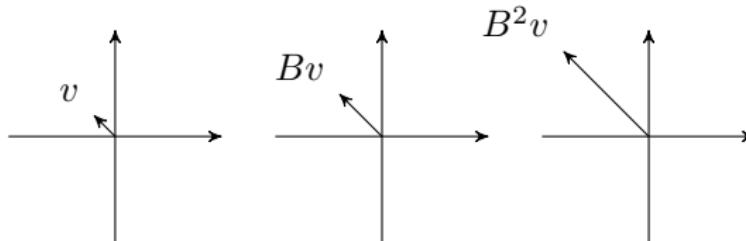
Setting

We repeatedly apply a symmetric matrix B to some vector $v \in \mathbb{R}^m$, i.e. we compute

$$Bv, \quad B(Bv) = B^2v, \quad B(B(Bv))) = B^3v, \quad \dots$$

How does v change?

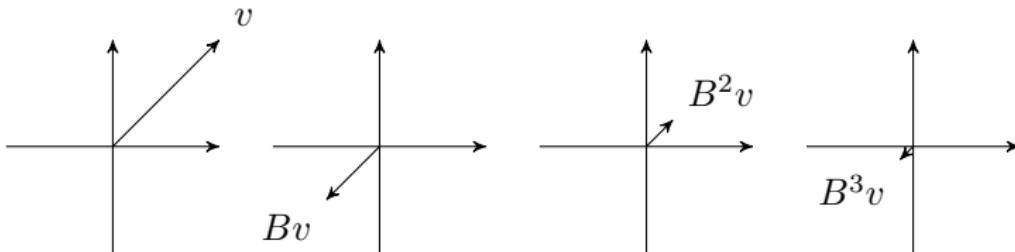
Example 1: v is an eigenvector with eigenvalue 2



The direction of v does not change, but its length doubles with each application of B .

Illustration

Example 2: v is an eigenvector with eigenvalue $-\frac{1}{2}$



For an arbitrary vector v

$$B^n v = \sum_{j=1}^m v_j^B \lambda_j^n \xi_j$$

- ▶ The weight λ_j^n grows most rapidly for eigenvalue with largest absolute value.
- ▶ Consequence:

The direction of $B^n v$ converges to the direction of the eigenvector with largest eigenvalue as n grows large.

Thanks to Sergio Bacallado and Peter Orbanz
for sharing the slides.

Lecture 3: Principle Component Analysis (PCA)

Reading: Section 14.5

GU4241/GR5241 Statistical Machine Learning

Linxí Liu
January 26, 2018

Quadratic Forms 二次型

In applications, symmetric matrices often occur in quadratic forms.

Definition

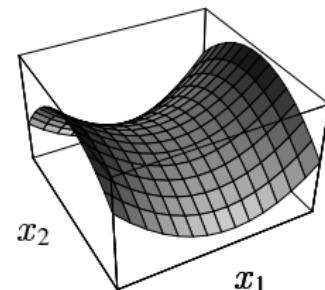
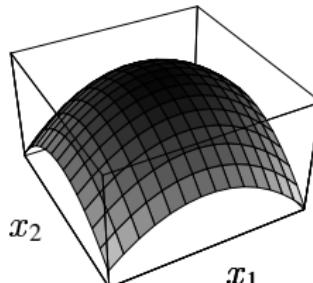
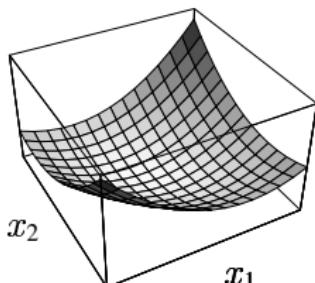
The **quadratic form** defined by a matrix A is the function

$$q_A : \mathbb{R}^m \rightarrow \mathbb{R}$$

$$x \mapsto \langle x, Ax \rangle$$

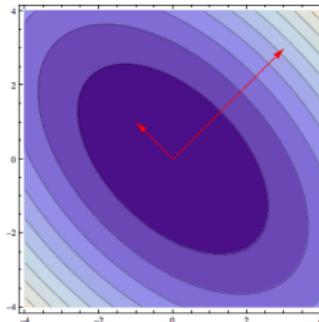
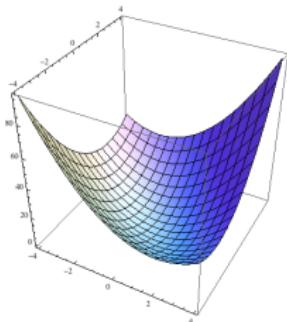
Intuition

A quadratic form is the m -dimensional analogue of a quadratic function ax^2 , with a vector substituted for the scalar x and the matrix A substituted for the scalar $a \in \mathbb{R}$.



Quadratic Forms

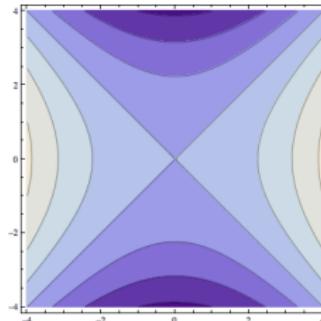
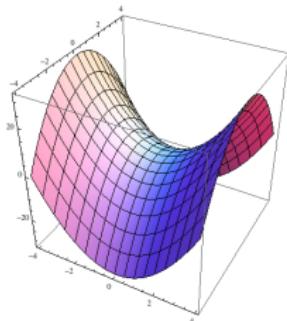
Here is the quadratic form for the matrix $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$:



- ▶ Left: The function value q_A is graphed on the vertical axis.
- ▶ Right: Each line corresponds to a constant function value of q_A . Dark color = small values.
- ▶ The red lines are eigenvector directions of A . Their lengths represent the (absolute) values of the eigenvalues.
- ▶ In this case, both eigenvalues are positive. If all eigenvalues are positive, the contours are ellipses. So:
positive definite matrices \leftrightarrow elliptic quadratic forms

Quadratic Forms

In this plot, the eigenvectors are axis-parallel, and one eigenvalue is negative:



The matrix here is $A = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$.

Intuition

- ▶ If we change the sign of one of the eigenvalue, the quadratic function along the corresponding eigen-axis flips.
- ▶ There is a point which is a minimum of the function along one axis direction, and a maximum along the other. Such a point is called a *saddle point*.

Application: Covariance Matrix

Recall: Covariance

The covariance of two random variables X_1, X_2 is

$$\text{Cov}[X_1, X_2] = \mathbb{E}[(X_1 - \mathbb{E}[X_1])(X_2 - \mathbb{E}[X_2])].$$

If $X_1 = X_2$, the covariance is the variance: $\text{Cov}[X, X] = \text{Var}[X]$.

Covariance matrix

If $X = (X_1, \dots, X_m)$ is a random vector with values in \mathbb{R}^m , the matrix of all covariances

$$\text{Cov}[X] := (\text{Cov}[X_i, X_j])_{i,j} = \begin{pmatrix} \text{Cov}[X_1, X_1] & \cdots & \text{Cov}[X_1, X_m] \\ \vdots & & \vdots \\ \text{Cov}[X_m, X_1] & \cdots & \text{Cov}[X_m, X_m] \end{pmatrix}$$

is called the **covariance matrix** of X .

协方差阵

Notation

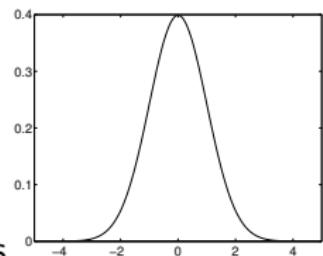
It is customary to denote the covariance matrix $\text{Cov}[X]$ by Σ .

Gaussian Distribution

Gaussian density in one dimension

$$p(x; \mu, \sigma) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

- ▶ μ = expected value of x , σ^2 = variance, σ = standard deviation
- ▶ The quotient $\frac{x-\mu}{\sigma}$ measures deviation of x from its expected value in units of σ (i.e. σ defines the length scale)



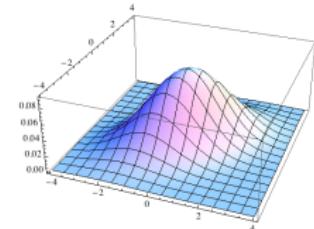
Gaussian density in m dimensions

The quadratic function

$$-\frac{(x - \mu)^2}{2\sigma^2} = -\frac{1}{2}(x - \mu)(\sigma^2)^{-1}(x - \mu)$$

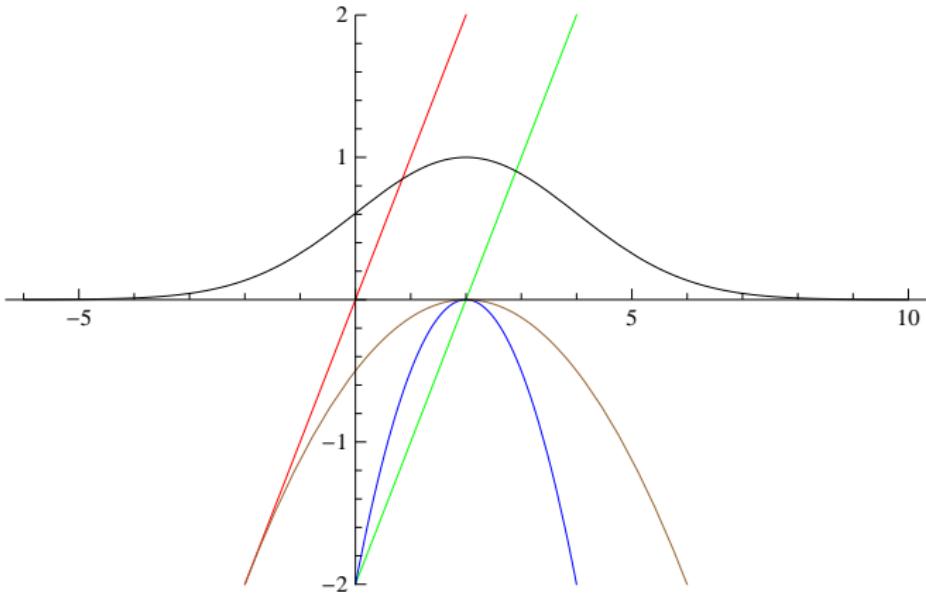
is replaced by a quadratic form:

$$p(\mathbf{x}; \boldsymbol{\mu}, \Sigma) := \frac{1}{\sqrt{2\pi \det(\Sigma)}} \exp\left(-\frac{1}{2} \langle (\mathbf{x} - \boldsymbol{\mu}), \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) \rangle\right)$$



Components of a 1D Gaussian

$$\mu = 2, \sigma = 2$$



- ▶ Red: $x \mapsto x$
- ▶ Green: $x \mapsto x - \mu$
- ▶ Blue: $x \mapsto -\frac{1}{2}(x - \mu)^2$

- ▶ Brown: $x \mapsto -\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2$
- ▶ Black: $x \mapsto \exp\left(-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right)$

Geometry of Gaussians

Covariance matrix of a Gaussian

If a random vector $X \in \mathbb{R}^m$ has Gaussian distribution with density $p(\mathbf{x}; \mu, \Sigma)$, its covariance matrix is $\text{Cov}[X] = \Sigma$. In other words, a Gaussian is parameterized by its covariance.

Observation

Since $\text{Cov}[X_i, X_j] = \text{Cov}[X_j, X_i]$, the covariance matrix is symmetric.

What is the eigenstructure of Σ ?

- We know: Σ symmetric \Rightarrow there is an eigenvector ONB
任何一个矩阵的特征向量都可以组成一组正交基
- Call the eigenvectors in this ONB ξ_1, \dots, ξ_m and their eigenvalues $\lambda_1, \dots, \lambda_m$
因为对称
- We can rotate the coordinate system to ξ_1, \dots, ξ_m . In the new coordinate system, Σ has the form
称序

$$\Sigma_{[\xi_1, \dots, \xi_n]} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{pmatrix} = \text{diag}(\lambda_1, \dots, \lambda_m)$$

Example

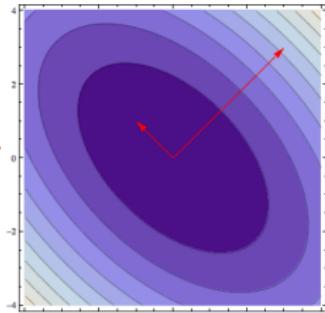
Quadratic form

$$\langle \mathbf{x}, \Sigma \mathbf{x} \rangle \quad \text{with} \quad \Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

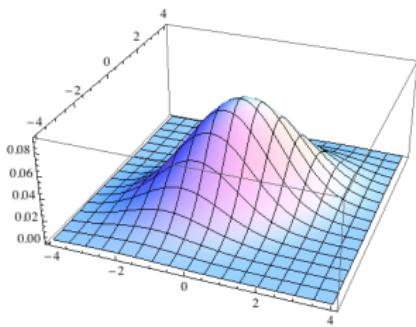
The eigenvectors are $(1, 1)$ and $(-1, 1)$ with eigenvalues 3 and 1.

Gaussian density

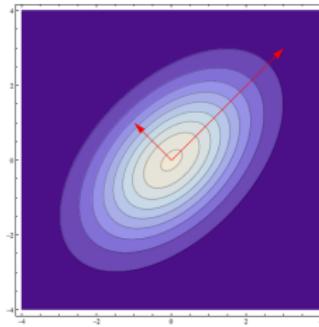
$p(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} = (0, 0)$.



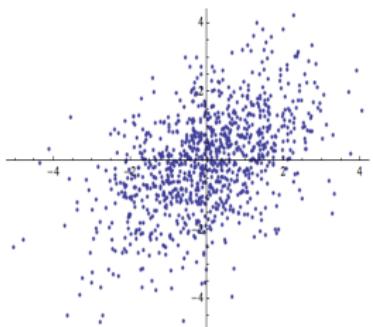
二次型下特征向量
总能指向根据变动最大的方向



Density graph



Density contour



1000 sample points 9 / 30

Interpretation

The ξ_i as random variables

Write e_1, \dots, e_m for the ONB of axis vectors. We can represent each ξ_i as

$$\xi_i = \sum_{j=1}^m \alpha_{ij} e_j$$

新政基向量的线性组合

Then $O = (\alpha_{ij})$ is the orthogonal transformation matrix between the two bases.

We can represent random vector $X \in \mathbb{R}^m$ sampled from the Gaussian in the eigen-ONB as

$$X_{[\xi_1, \dots, \xi_m]} = (X'_1, \dots, X'_m) \quad \text{with} \quad X'_i = \sum_{j=1}^m \alpha_{ij} X_j$$

Since the X_j are random variables (and the α_{ij} are fixed), each X'_i is a scalar random variable.

Interpretation

Meaning of the random variables ξ_i

For any Gaussian $p(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$, we can

1. shift the origin of the coordinate system into $\boldsymbol{\mu}$
2. rotate the coordinate system to the eigen-ONB of Σ .

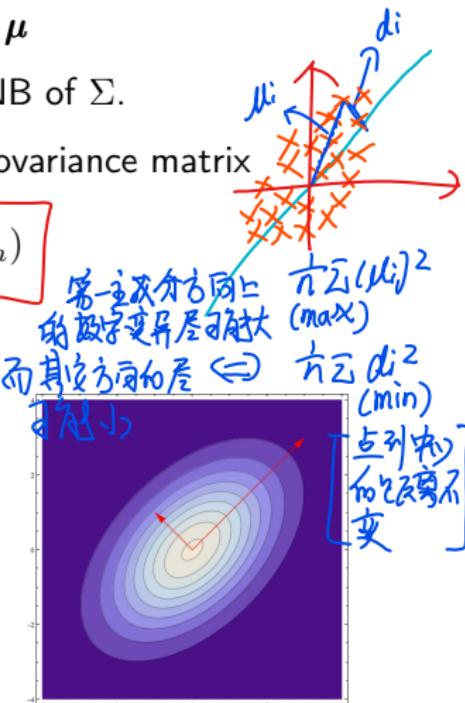
In this new coordinate system, the Gaussian has covariance matrix

$$\Sigma_{[\xi_1, \dots, \xi_m]} = \text{diag}(\lambda_1, \dots, \lambda_m)$$

where λ_i are the eigenvalues of Σ .

Gaussian in the new coordinates

A Gaussian vector $X_{[\xi_1, \dots, \xi_m]}$ represented in the new coordinates consists of m independent 1D Gaussian variables X'_i . Each X'_i has mean 0 and variance λ_i .



$$Z(X) = \mu$$

Principal Component Analysis

$$\bar{Z} = \begin{pmatrix} \bar{b}_{11} & \bar{b}_{12} & \dots & \bar{b}_{1P} \\ \vdots & \ddots & \ddots & \vdots \\ \bar{b}_{P1} & \dots & \dots & \bar{b}_{PP} \end{pmatrix}$$

标准化之前

(利用 2)

$$\text{每个主成分是 } \left\{ \begin{array}{l} Z_1 = a_1^T X \\ Z_2 = a_2^T X \\ \vdots \\ Z_P = a_P^T X \end{array} \right.$$

$$\text{原 Base 的 } \left\{ \begin{array}{l} Z_1 = a_1^T X \\ Z_2 = a_2^T X \\ \vdots \\ Z_P = a_P^T X \end{array} \right.$$

$$\text{线性组合 } \left\{ \begin{array}{l} Z_1 = a_1^T X \\ Z_2 = a_2^T X \\ \vdots \\ Z_P = a_P^T X \end{array} \right.$$

$$Z_P = a_P^T X$$

$$\text{Var}(Z_i) = a_i^T \bar{Z}^T \bar{Z} a_i$$

$$\text{Cov}(Z_i, Z_j) = a_i^T \bar{Z}^T \bar{Z} a_j$$

目标找到 a_1 使

$$\text{Var}(Z_1) \text{ 最大}$$

- This is the most popular unsupervised procedure ever.
- Invented by Karl Pearson (1901).
- Developed by Harold Hotelling (1933).
- What does it do? It provides a way to visualize high dimensional data, summarizing the most important information. 又： Z 是对称阵 \Rightarrow 只有特征值

$$Q = [a_1, a_2, \dots, a_P] \Rightarrow Q^T \bar{Z} Q = \Lambda = [\lambda_1, \dots, \lambda_P]$$

Z 的特点： $\left\{ \begin{array}{l} Z(X) = \bar{E}(Q^T X) = Q^T \mu \\ \text{Var}(Z) = \text{Var}(Q^T X) = Q^T Q = \Lambda \end{array} \right.$

$$\left\{ \begin{array}{l} \text{tr}(\Lambda) = \text{tr}(Q^T \bar{Z} Q) = \text{tr}(\bar{Z}) \Rightarrow \sum_{i=1}^P \lambda_i = \sum_{i=1}^P b_{ii} \\ \text{tr}(\Lambda) = \text{tr}(Q^T Q) = \text{tr}(Q) \Rightarrow \sum_{i=1}^P \text{Var}(Z_i) = \sum_{i=1}^P \text{Var}(X_i) \end{array} \right.$$

$$a_1^T \bar{Z} a_1 = \lambda_1$$

$$a_1^T \bar{Z} a_1 = a_1 \lambda_1$$

$$\Rightarrow \underbrace{\bar{Z} a_1}_{\downarrow} = \underbrace{a_1}_{\downarrow} \lambda_1$$

$$\underbrace{\bar{Z} a_1}_{\downarrow} = \underbrace{a_1}_{\downarrow} \lambda_1$$

特征向量 特征值

$\frac{\lambda_i}{\sum \lambda_i} \Rightarrow$ 叫做贡献率

标准化后：利用 R(相关系数)

$$X = QZ \Rightarrow X_j = q_{j1}Z_1 + q_{j2}Z_2 + \dots + q_{jp}Z_p$$

$$\text{cov}(X_j, Z_i) = q_{ji} \lambda_i$$

$$P(X_j, Z_i) = \frac{\sqrt{\lambda_i}}{\sqrt{b_{ii}}} q_{ji}$$

$$b_{ij} = q_{j1}^2 \lambda_1 + q_{j2}^2 \lambda_2 + \dots + q_{jp}^2 \lambda_p$$

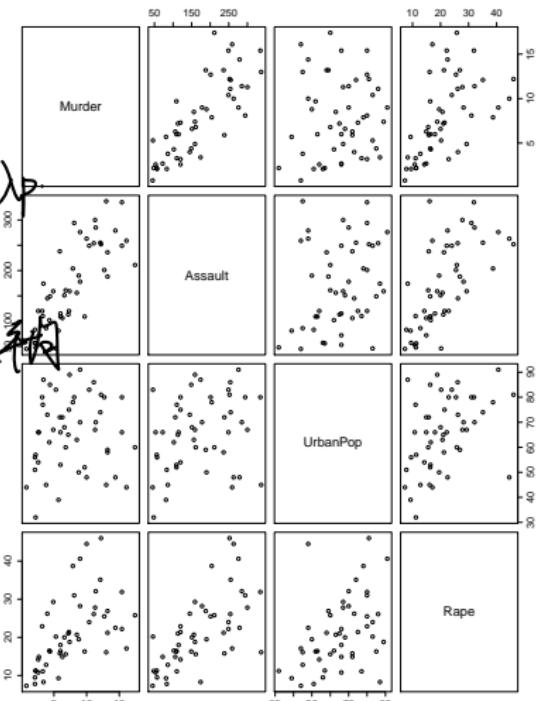
$$q_{j1}^2 + \dots + q_{jp}^2 = 1$$

故 b_{ij} 是 $\lambda_1, \dots, \lambda_p$ 的加权平均数

$$P_j | 1 \dots p = \sum_{i=1}^p P(X_j, Z_i)$$

$$= \sum_{i=1}^p \frac{\lambda_i q_{ji}^2}{b_{ii}} = 1$$

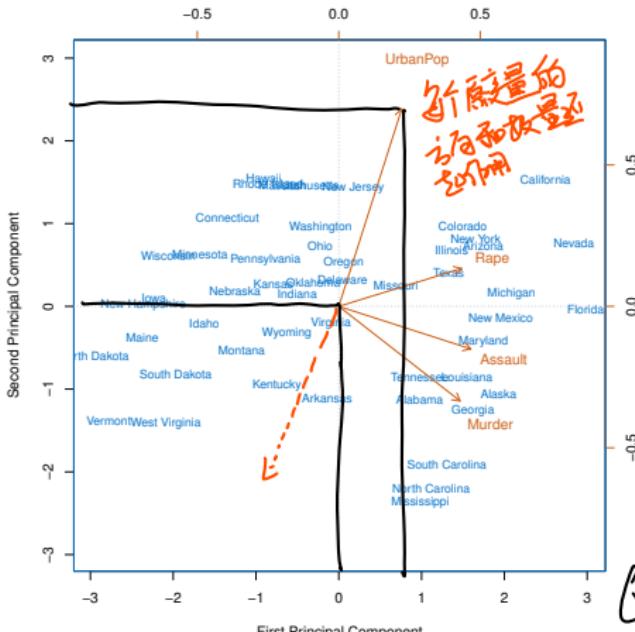
所以每个变量对均为进行了
一削弱的解释，加起来为1。



$$\begin{cases} E(Z) = 0 & b_{ii} = 1 \\ \sum \lambda_i = p & \text{恒成立} \end{cases}$$

$$P(X_j, Z_i) = \sqrt{\lambda_i} q_{ji}$$

What is PCA good for?



① 行主成分的特征组合向量成了新变量

$$\text{如: } \text{UrbanPop} = 2.5Z_1 + 0.8Z_2$$

② 行主成分是之前变量的线性组合.

$$Z = Q^T X$$

$$X = Q Z$$

$$Q = [q_1, \dots, q_p]$$

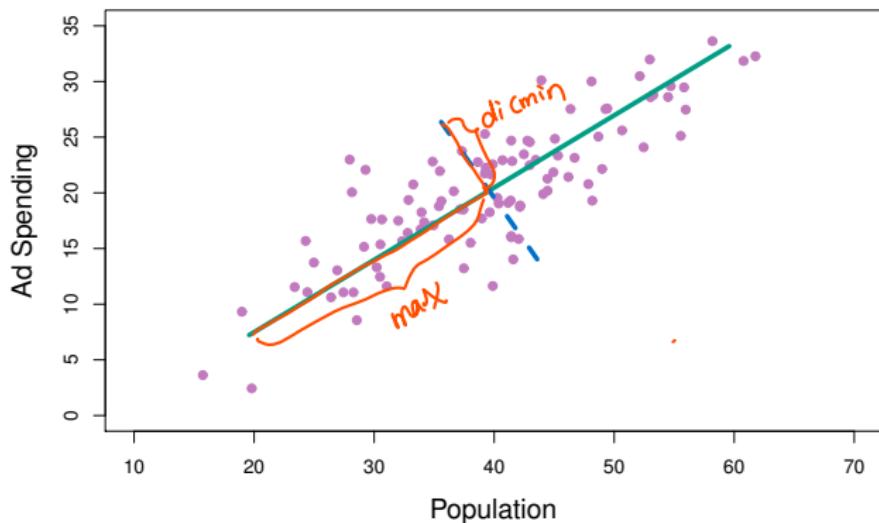
Q是外加的
组合成X

Q是又怎
组合成X

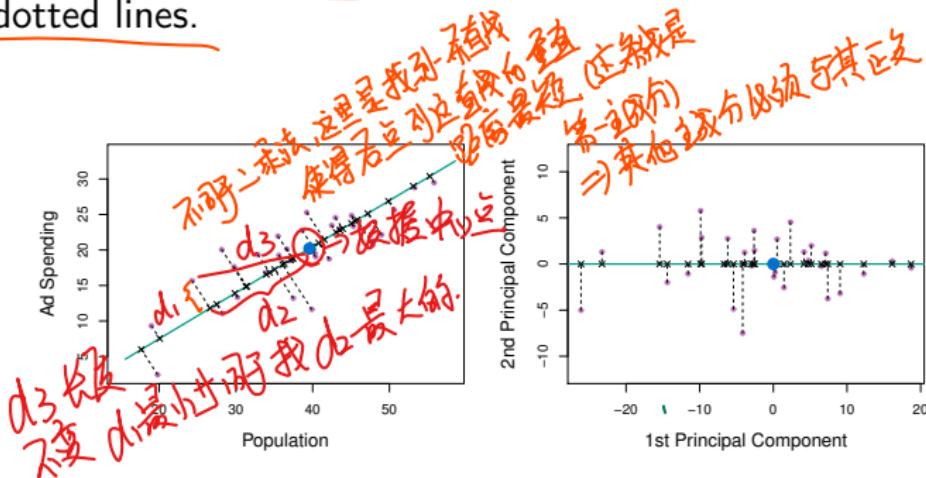
ISL Figure 10.1

What is the first principal component?

It is the vector which passes the closest to a cloud of samples, in terms of squared Euclidean distance.



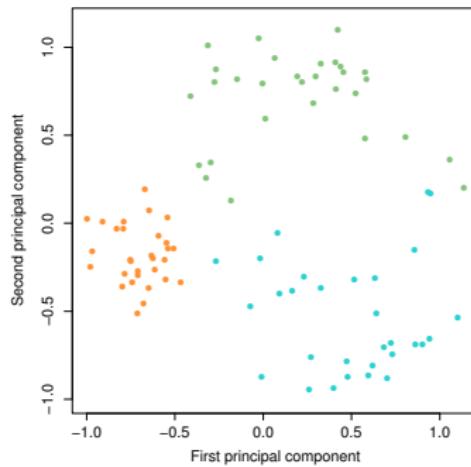
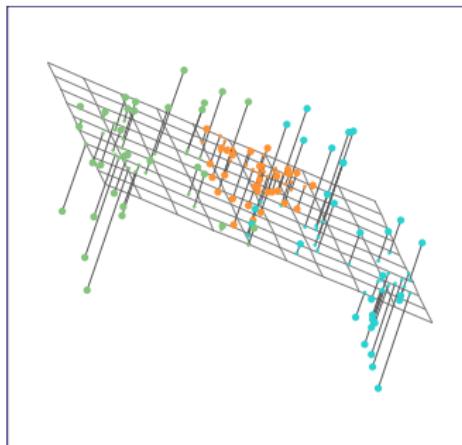
i.e. The green direction minimizes the average squared length of the dotted lines.



ISL Figure 6.15

What does this look like with 3 variables?

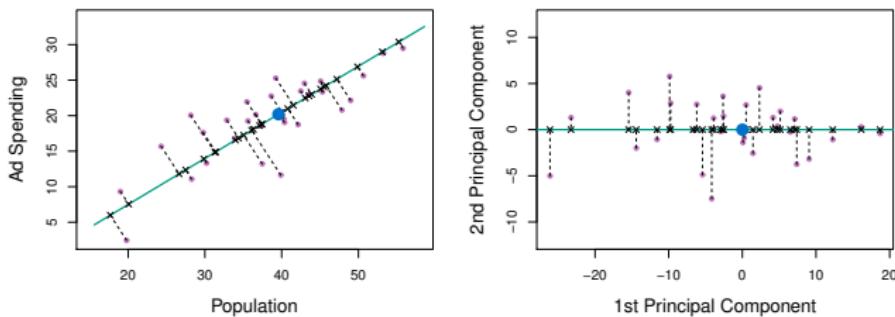
The first two principal components span a plane which is closest to the data.



ISL Figure 10.2

A second interpretation

The projection onto the first principal component is the one with the highest variance.

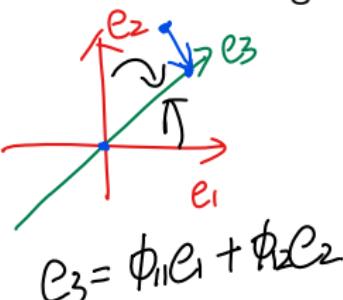


ISL Figure 6.15

How do we say this in math?

Let \mathbf{X} be a data matrix with n samples, and p variables. From each variable, we subtract the mean of the column; i.e. we center the variables.

To find the first principal component $\phi_1 = (\phi_{11}, \dots, \phi_{p1})$, we solve the following optimization



改变量后的新生代

$$\max_{\phi_{11}, \dots, \phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \right\} \quad \max \alpha_i^T X \alpha_i \\ \alpha_i^T \alpha_i = 1$$

subject to $\sum_{j=1}^p \phi_{j1}^2 = 1$.

Projection of the i th sample onto ϕ_1 . Also known as the score z_{i1}

How do we say this in math?

Let \mathbf{X} be a data matrix with n samples, and p variables. From each variable, we subtract the mean of the column; i.e. we **center** the variables.

To find the first principal component $\phi_1 = (\phi_{11}, \dots, \phi_{p1})$, we solve the following optimization

$$\max_{\phi_{11}, \dots, \phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \right\}$$

subject to $\sum_{j=1}^p \phi_{j1}^2 = 1.$

Variance of the n samples projected onto ϕ_1 .

How do we say this in math?

To find the second principal component $\phi_2 = (\phi_{12}, \dots, \phi_{p2})$, we solve the following optimization

$$\max_{\phi_{12}, \dots, \phi_{p2}} \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j2} x_{ij} \right)^2 \right\}$$

subject to $\sum_{j=1}^p \phi_{j2}^2 = 1$ and $\sum_{j=1}^p \phi_{j1} \phi_{j2} = 0$.

$a_i^\top a_j = 0$

First and second principal components must be orthogonal.

How do we say this in math?

To find the second principal component $\phi_2 = (\phi_{12}, \dots, \phi_{p2})$, we solve the following optimization

$$\max_{\phi_{12}, \dots, \phi_{p2}} \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j2} x_{ij} \right)^2 \right\}$$

subject to $\sum_{j=1}^p \phi_{j2}^2 = 1$ and $\sum_{j=1}^p \phi_{j1} \phi_{j2} = 0$.

First and second principal components must be orthogonal.

Equivalent to saying that the scores (z_{11}, \dots, z_{n1}) and (z_{12}, \dots, z_{n2}) are uncorrelated.

(7)

Solving the optimization

This optimization is fundamental in linear algebra. It is satisfied by either:

- The singular value decomposition (SVD) of \mathbf{X} :

$$u_i = \langle \mathbf{x}, \phi_i \rangle$$

$$u_1, \dots, u_n$$

$$\mathbf{X} = \mathbf{U}\Sigma\Phi^T$$

$$\sum u_i^2 = (\mathbf{x}\phi_i)^T(\mathbf{x}\phi_i)$$

$= \phi_i^T \mathbf{x}^T \mathbf{x} \phi_i$
 $= \phi_i^T V \Sigma U^T \phi_i$
 $= \alpha_i^T \Sigma \alpha_i$
 $= \sum_{i=1}^n \alpha_i^2 \lambda_i$

where the ith column of Φ is the ith principal component ϕ_i ,
and the ith column of $\mathbf{U}\Sigma$ is the ith vector of scores
 (z_{1i}, \dots, z_{ni}) .

$\alpha_1 = 1$
 $\alpha_2 = \alpha_3 = \dots = \alpha_p = 0$

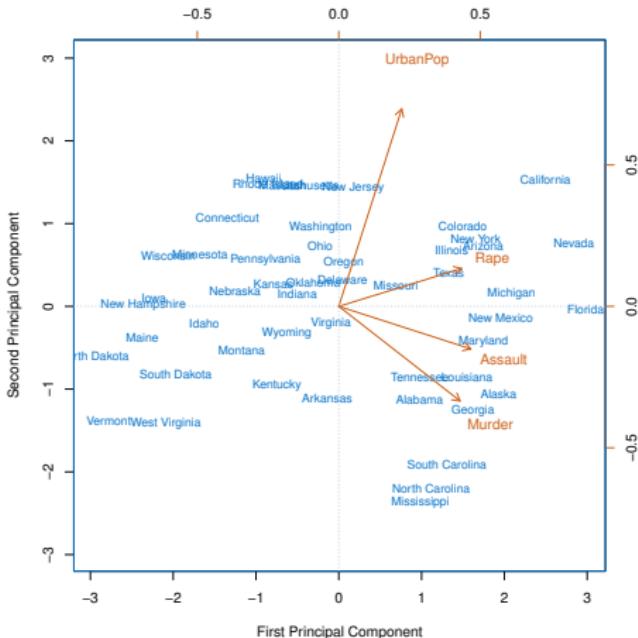
- The eigendecomposition of $\mathbf{X}^T \mathbf{X}$:

$$\boxed{\begin{aligned} \Sigma &= (\lambda_1 \dots \lambda_n) \\ \mathbf{X}^T \mathbf{X} &= V \Sigma V^T \end{aligned}}$$

$$\mathbf{X}\vec{\phi}_i = \begin{pmatrix} \langle \mathbf{x}_1 | \vec{\phi}_i \rangle \\ \vdots \\ \langle \mathbf{x}_n | \vec{\phi}_i \rangle \end{pmatrix} \quad \mathbf{X}^T \mathbf{X} = \Phi \Sigma^2 \Phi^T$$

after centered
 $\mathbf{X}^T \mathbf{X}$ is correlation matrix

PCA in practice: The biplot



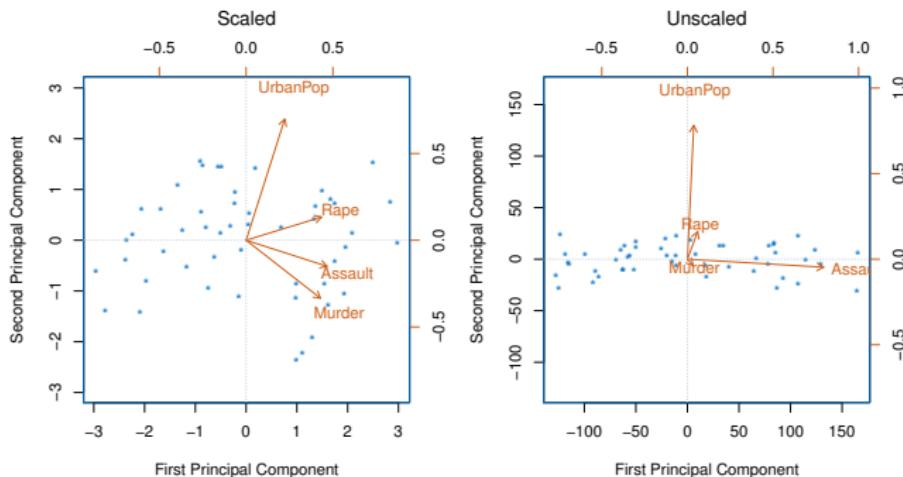
ISL Figure 10.1

Scaling the variables

Most of the time, we don't care about the absolute numerical value of a variable. We care about the value relative to the spread observed in the sample.

Before PCA, in addition to centering each variable, we also multiply it times a constant to make its variance equal to 1. 标准化

Example: scaled vs. unscaled PCA



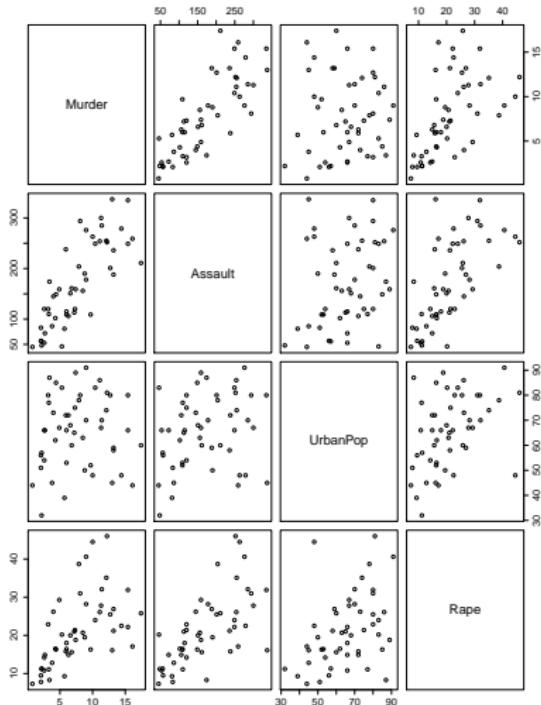
ISL Figure 10.3

Scaling the variables

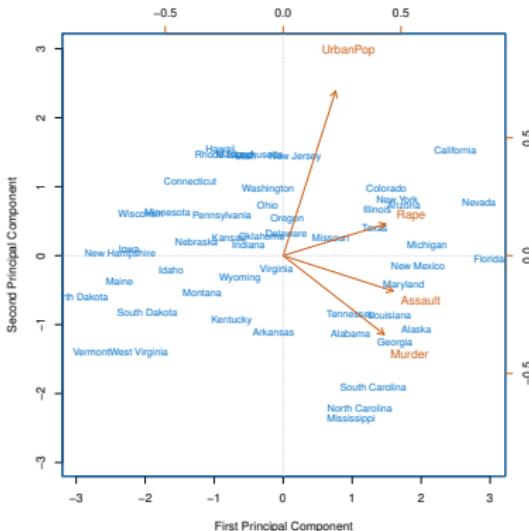
In special cases, we have variables measured in the same unit; e.g. gene expression levels for different genes.

Therefore, we care about the absolute value of the variables and we can perform PCA without scaling.

How many principal components are enough?



How many principal components are enough?



We said 2 principal components capture most of the relevant information. But how can we tell?

The proportion of variance explained

We can think of the top **principal components** as directions in space in which the data vary the most.

The i th **score vector** (z_{1i}, \dots, z_{ni}) can be interpreted as a *new* variable. The variance of this variable decreases as we take i from 1 to p . However, the total variance of the score vectors is the same as the total variance of the original variables:

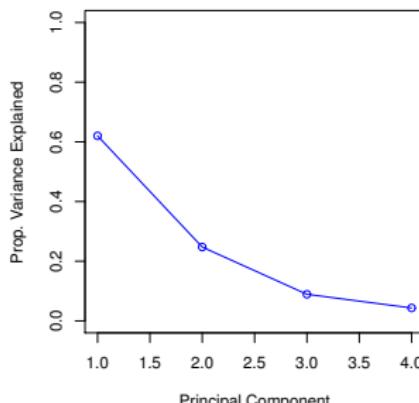
$$\sum_{i=1}^p \frac{1}{n} \sum_{j=1}^n z_{ji}^2 = \sum_{k=1}^p \text{Var}(x_k).$$

We can quantify how much of the variance is captured by the first m principal components/score variables.

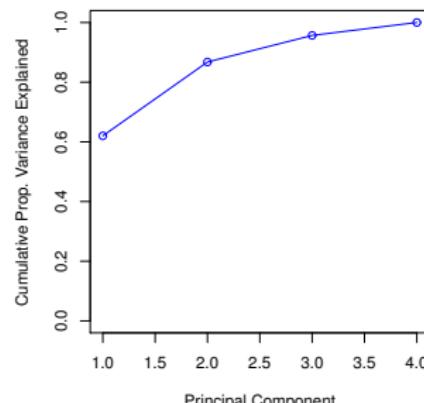
The proportion of variance explained

The variance of the m th score variable is:

$$\frac{1}{n} \sum_{i=1}^n z_{im}^2 = \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{jm} x_{ij} \right)^2 = \frac{1}{n} \Sigma_{mm}^2.$$



Scree plot



Generalizations of PCA

PCA works under a Euclidean geometry in the space of variables.
Often, the natural geometry is different:

- ▶ We expect some variables to be “closer” to each other than to other variables.
- ▶ Some correlations between variables would be more surprising than others.

Examples:

- ▶ Variables are pixel values, samples are different images of the brain. We expect neighboring pixels to have stronger correlations.
- ▶ Variables are rainfall measurements at different regions. We expect neighboring regions to have higher correlations.

Generalizations of PCA

There are ways to include this knowledge in a PCA. See:

1. Susan Holmes. *Multivariate Analysis, the French way*. (2006).
2. Omar de la Cruz and Susan Holmes. *An introduction to the duality diagram*. (2011).
3. Stéphane Dray and Thibaut Jombart. *Revisiting Guerry's data: Introducing spatial constraints in multivariate analysis*. (2011).
4. Genevera Allen, Logan Grosenick, and Jonathan Taylor. *A Generalized Least Squares Matrix Decomposition*. (2011).

Thanks to Sergio Bacallado and Peter Orbanz
for sharing the slides.

Lecture 4: K-means and K-nearest neighbors

Reading: Sections 13.3, 14.3.6

GU4241/GR5241 Statistical Machine Learning

Linxí Liu
January 26, 2018

Clustering

We assign a class to each sample in the data matrix. However, the class *is not an output variable*; we only use input variables.

Clustering is an **unsupervised** procedure, whose goal is to find homogeneous subgroups among the observations. It has wide applications in practice. **Image segmentation, handwritten digit identification, vector quantization**

We will discuss 4 algorithms in this semester:

- ▶ K -means clustering
- ▶ K -medoids clustering
- ▶ Hierarchical clustering
- ▶ EM algorithm

Handwritten digit identification



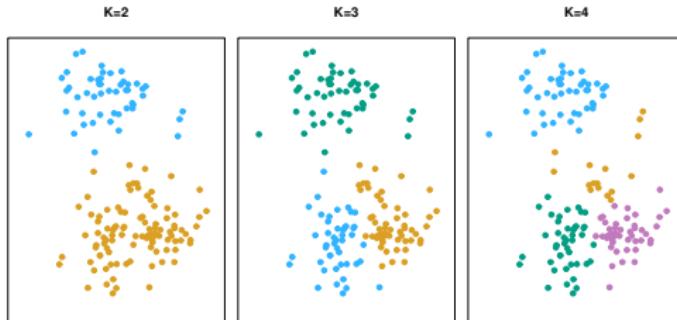
FIGURE 11.9. Examples of training cases from ZIP code data. Each image is a 16×16 8-bit grayscale representation of a handwritten digit.

Image segmentation



K-means clustering

- K is the number of clusters and must be fixed in advance.



ISL Figure 10.5

- The goal of this method is to maximize the similarity of samples within each cluster:

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

K-means clustering algorithm

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.
2. Iterate these two steps until the clustering is constant:
 - ▶ Find the *centroid* of each cluster ℓ ; i.e. the average $\bar{x}_{\ell,:}$ of all the samples in the cluster:
$$\bar{x}_{\ell,j} = \frac{1}{|\{i : C(i) = \ell\}|} \sum_{i:C(i)=\ell} x_{i,j} \quad \text{for } j = 1, \dots, p.$$
 - ▶ Reassign each sample to the nearest centroid.

K-means clustering algorithm

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 14

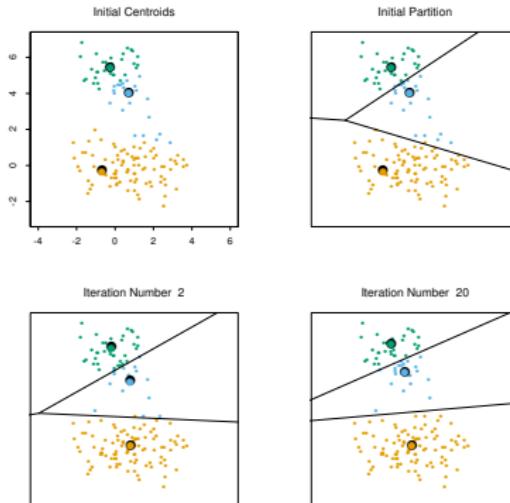


FIGURE 14.6. Successive iterations of the K -means clustering algorithm for the simulated data of Figure 14.4.

Properties of K -means clustering

- ▶ The algorithm always converges to a local minimum of

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

Why? When d is the Euclidean distance

$$\frac{1}{2} \sum_{C(i)=\ell} \sum_{C(j)=\ell} d(x_i, x_j) = |N_\ell| \sum_{C(i)=\ell} d(x_i, \bar{x}_\ell)$$

Properties of K -means clustering

- ▶ The algorithm always converges to a local minimum of

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

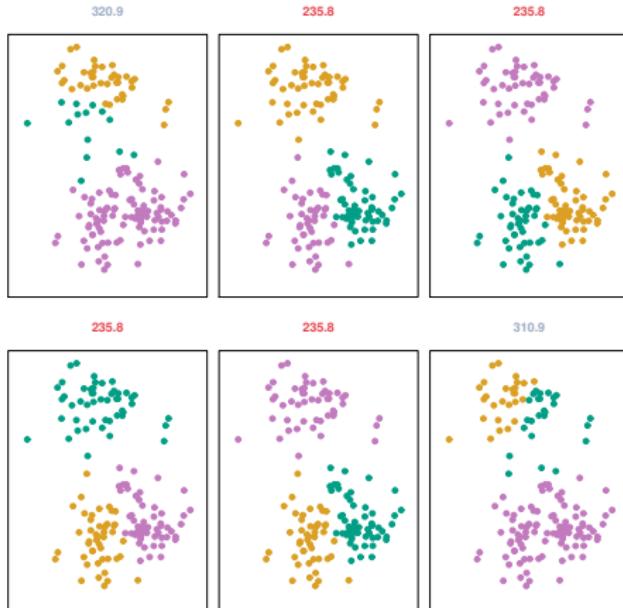
Why? When d is the Euclidean distance

$$\frac{1}{2} \sum_{C(i)=\ell} \sum_{C(j)=\ell} d(x_i, x_j) = |N_\ell| \sum_{C(i)=\ell} d(x_i, \bar{x}_\ell)$$

This side can only be reduced in each iteration.

- ▶ Each initialization could yield a different minimum.

Example: K -means output with different initializations



In practice, we start from many random initializations and choose the output which minimizes the objective function.

ISL Figure 10.7

Practical Issues

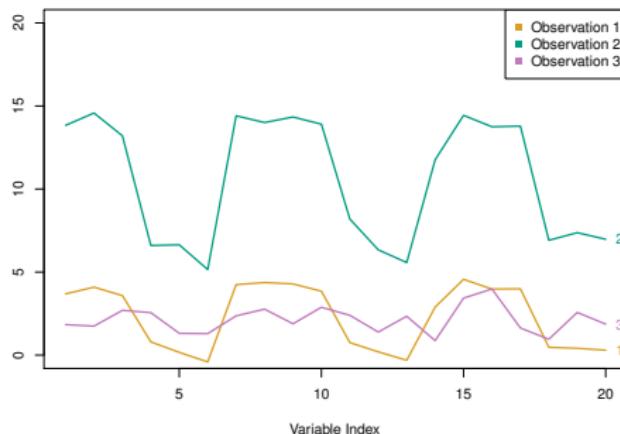
- ▶ Categorical features are usually coded as dummy variables:

$$X = 1, 2, \text{ or } 3 \rightarrow \begin{matrix} (1 \ 0 \ 0) \\ (0 \ 1 \ 0) \\ (0 \ 0 \ 1) \end{matrix}$$

- ▶ Weighting is also possible
- ▶ How to choose the number of clusters K ?

Correlation distance

- ▶ Euclidean distance would cluster all customers who purchase few things (orange and purple).
- ▶ Perhaps we want to cluster customers who purchase *similar* things (orange and teal).
- ▶ Then, the **correlation distance** may be a more appropriate measure of dissimilarity between samples.



Fact of correlation distance

Correlation is defined by

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}},$$

where \bar{x}_i = mean of obeservation i .

If observations are standardized:

$$x_{ij} \leftarrow \frac{x_{ij} - \bar{x}_i}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2}},$$

then $2(1 - \rho(x_i, x_{i'})) = \sum_j (x_{ij} - x_{i'j})^2$.

K-medoids clustering

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.
2. Iterate these two steps until the clustering is constant:
 - ▶ For a given cluster assignment C find the **observation** in the cluster minimizing total pairwise distance with the other cluster members:

$$i_k^* = \operatorname{argmin}_{\{i: C(i)=k\}} \sum_{C(i')=k} d(x_i, x_{i'}).$$

Then $z_k = x_{i_k^*}$, $k = 1, 2, \dots, K$ are the current estimates of the cluster centers.

- ▶ Given a current set of cluster centers $\{z_1, \dots, z_K\}$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \operatorname{argmin}_{1 \leq k \leq K} d(x_i, z_k).$$

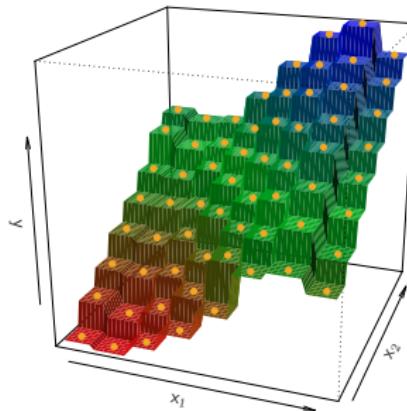
K-medoids clustering

- ▶ Same as *K*-means, except that centroid is required to be one of the observations.
- ▶ Advantage: centroid is one of the observations— useful, for example when features are 0 or 1. Also, one only needs pairwise distances for *K*-medoids rather than the raw observations.

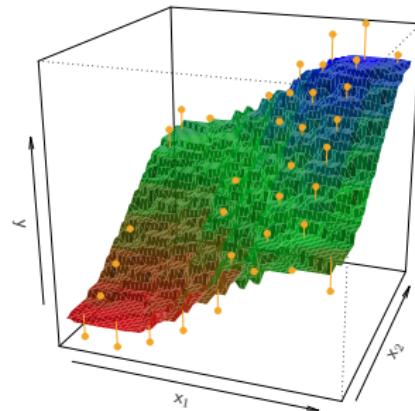
K-nearest neighbors regression

KNN regression: prototypical nonparametric method.
Given a training set (\mathbf{X}, \mathbf{y}) :

$$\hat{f}(x) = \frac{1}{K} \sum_{i \in N_K(x)} y_i$$

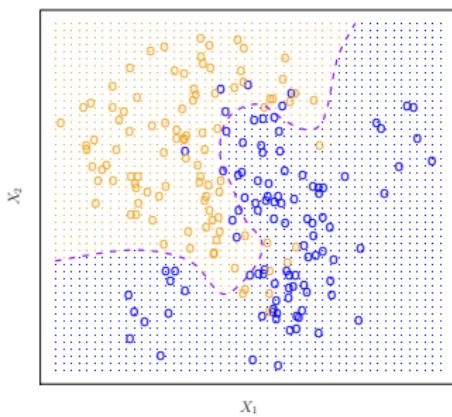


$$K = 1$$



$$K = 9$$

Classification problem



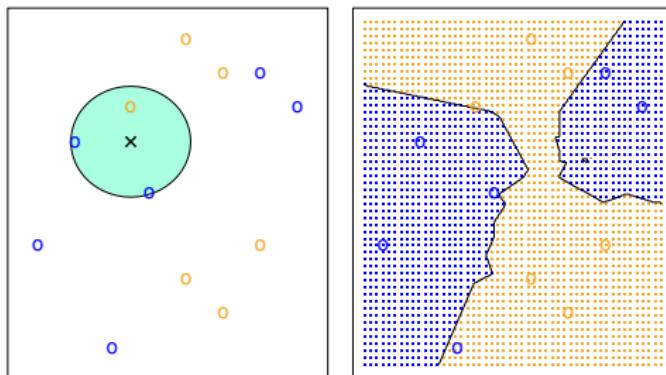
ISL Figure 2.13

Recall:

- ▶ $X = (X_1, X_2)$ are inputs.
- ▶ Color $Y \in \{\text{Yellow , Blue}\}$ is the output.
- ▶ (X, Y) have a joint distribution.
- ▶ Purple line is *Bayes boundary* — the best we could do if we knew the joint distribution of (X, Y)

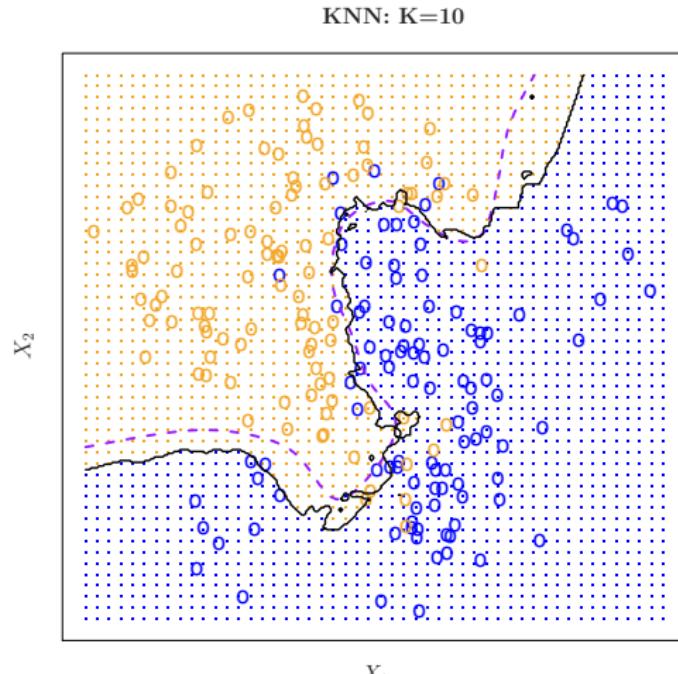
K-nearest neighbors

To assign a color to the input \times , we look at its $K = 3$ nearest neighbors. We predict the color of the majority of the neighbors.



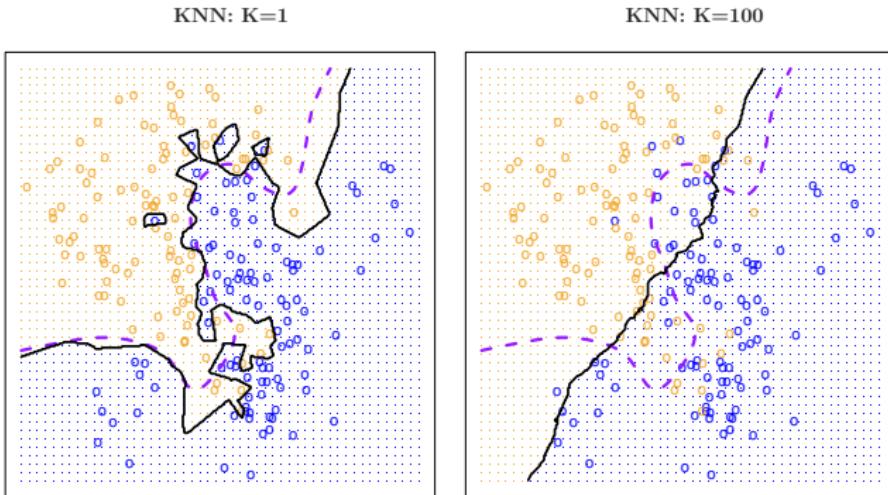
ISL Figure 2.14

K-nearest neighbors also has a decision boundary



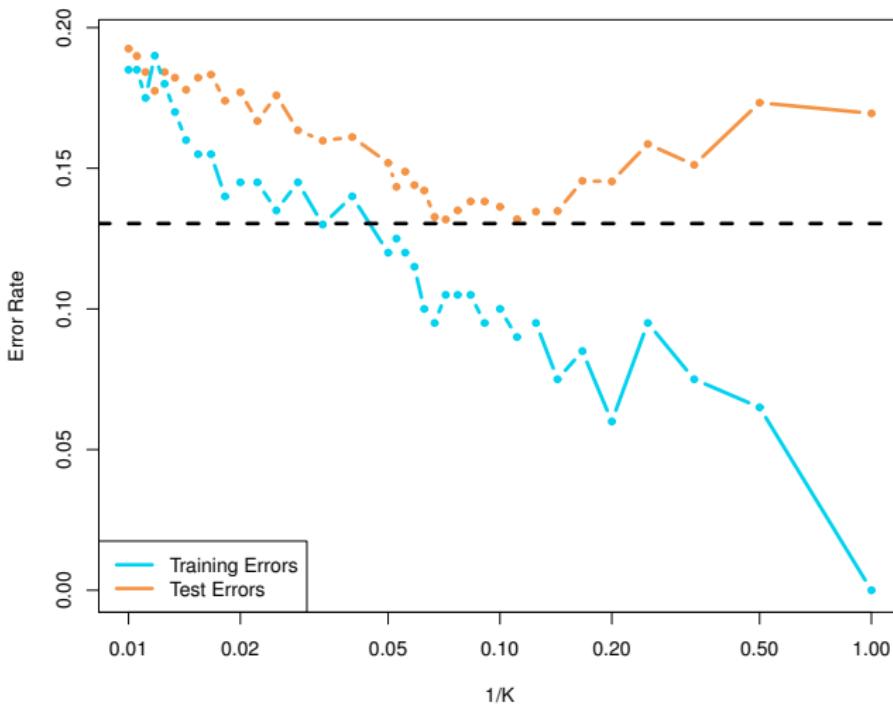
ISL Figure 2.15

The higher K , the smoother the decision boundary



ISL Figure 2.16

Test error vs. training error



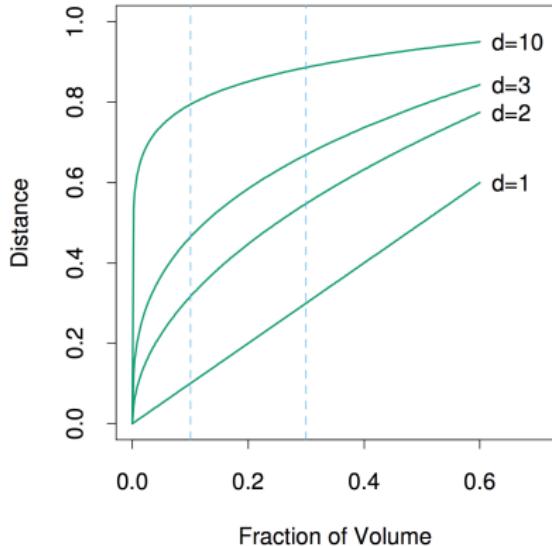
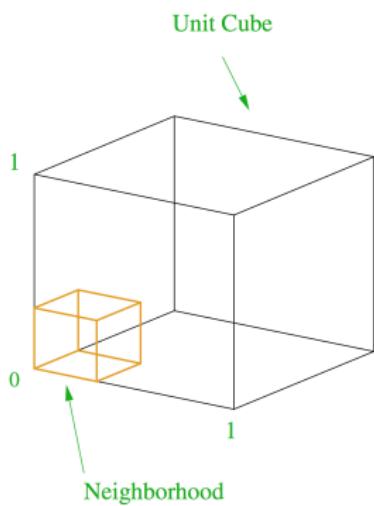
ISL Figure 2.17

Curse of dimensionality

K-nearest neighbors can fail in high dimensions, because it becomes difficult to gather K observations close to a target point x_0 :

- ▶ near neighborhoods tend to be spatially large, the estimates are biased.
- ▶ reducing the spatial size of the neighborhood means reducing K , and the variance of the estimate increases.

Curse of dimensionality



ESL Figure 2.6

- ▶ We want to obtain a hypercubical neighborhood about a target point to capture a fraction r of the observations.
- ▶ The expected edge length will be $e_p(r) = r^{1/p}$. In ten dimensions, $e_{10}(0.01) = 63\%$.

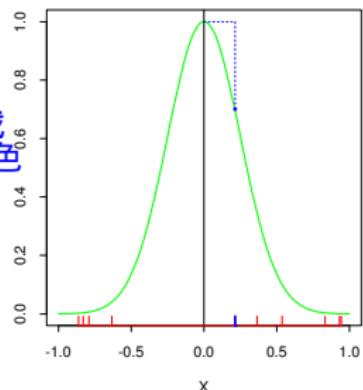
Example

- ▶ 1000 training examples x_i generated uniformly on $[-1, 1]^p$.
- ▶ $Y = f(X) = e^{-8\|X\|^2}$ (no measurement error).
- ▶ use the 1-nearest-neighbor rule to predict y_0 at the test-point $x_0 = 0$.

$$\begin{aligned}\text{MSE}(x_0) &= \mathbb{E}_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\ &= \mathbb{E}_{\mathcal{T}}[\hat{y}_0 - \mathbb{E}_{\mathcal{T}}(\hat{y}_0)]^2 + [\mathbb{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\ &= \text{Var}_{\mathcal{T}}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0).\end{aligned}$$

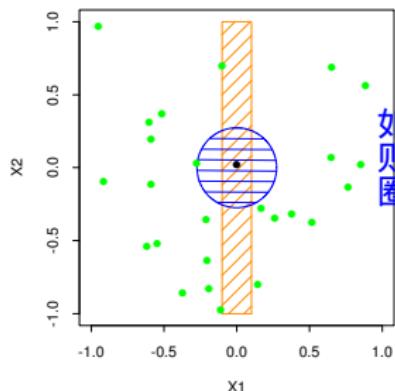
1-NN in One Dimension

打算计算 $x=0$ 处的y值，寻找最近的点为蓝色的那个点



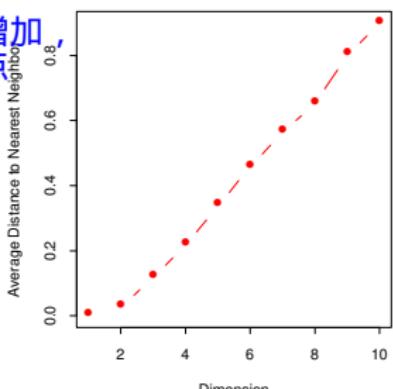
1-NN in One vs. Two Dimensions

如果维度变成二维，则最近的那个点为圆圈范围内的那个点

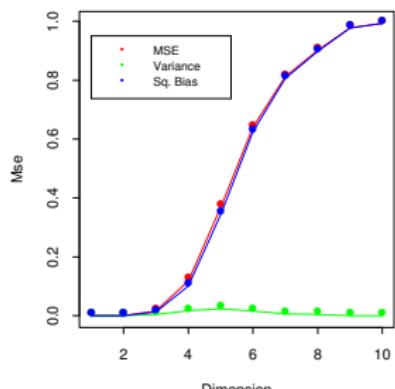


Distance to 1-NN vs. Dimension

随着维度的增加，最近的那个点的距离也在增加



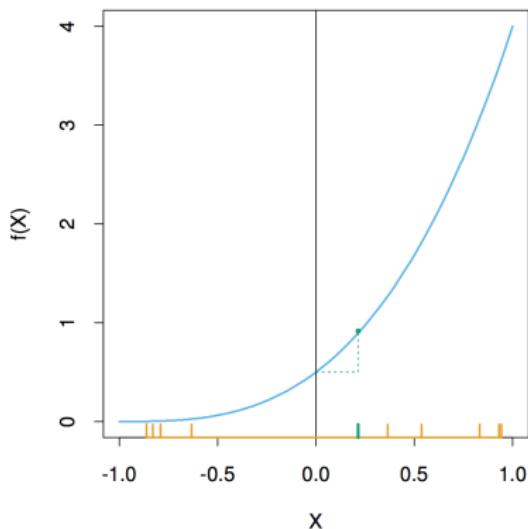
MSE vs. Dimension



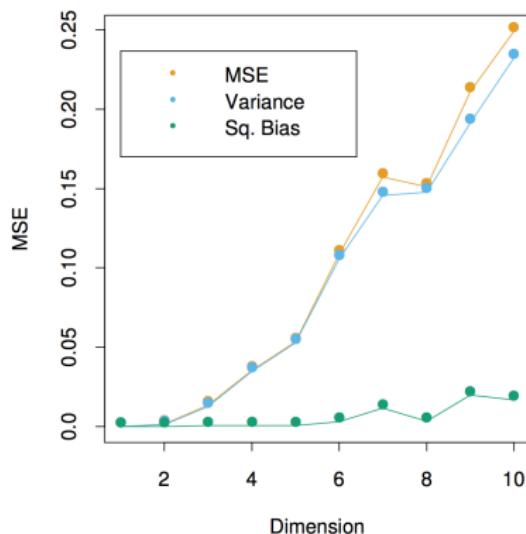
An example when the variance dominates

Assume the regression function is: $f(X) = \frac{1}{2}(X_1 + 1)^3$.

1-NN in One Dimension



MSE vs. Dimension



Lecture 6: Linear Regression

Reading: Sections 3.2, 3.3

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
February 2, 2018

Multiple linear regression

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

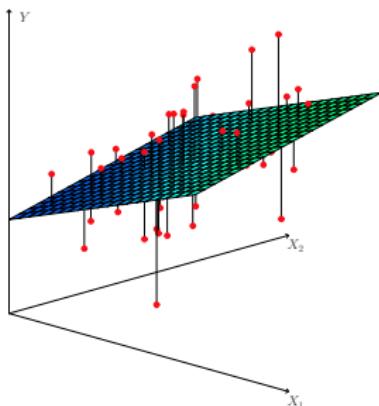


Figure 3.4

$$\varepsilon \sim \mathcal{N}(0, \sigma) \quad \text{i.i.d.}$$

or, in matrix notation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon,$$

where $\mathbf{y} = (y_1, \dots, y_n)^T$,
 $\boldsymbol{\beta} = (\beta_0, \dots, \beta_p)^T$ and \mathbf{X} is our usual data matrix with an extra column of ones on the left to account for the intercept.

The estimates $\hat{\beta}$

Our goal is to minimize the RSS (residual sum of squares, training error):

$$\begin{aligned}\text{RSS} &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i,1} - \cdots - \beta_p x_{i,p})^2.\end{aligned}$$

This is minimized by the vector $\hat{\beta}$:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

This only exists when $\mathbf{X}^T \mathbf{X}$ is invertible. This requires $n \geq p$.

Multiple linear regression answers several questions

- ▶ Is at least one of the variables X_i useful for predicting the outcome Y ? **F test**
- ▶ Which subset of the predictors is most important?
- ▶ How good is a linear model for these data?
- ▶ Given a set of predictor values, what is a likely value for Y , and how accurate is this prediction?

Testing whether a group of variables is important

- ▶ F-test:

$$H_0 : \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0.$$

RSS₀ is the residual sum of squares for the model in H_0 .

$$F = \frac{(\text{RSS}_0 - \text{RSS})/q}{\text{RSS}/(n-p-1)}.$$

q 表示的是 reduced model 的变量的个数

- ▶ Special case: $q = p$. Test whether any of the predictors are important.
- ▶ Special case: $q = 1$, exclude a single variable. **Test whether this variable is important $\sim t$ -tests in R output.** **Not be careful with multiple testing.**

t test 体现的是一种 marginal effect

Which subset of variables are important?

When choosing a subset of the predictors, we have 2^p choices. We cannot test every possible subset!

Instead we will use a **stepwise approach**:

1. Construct a sequence of p models with increasing number of variables.
2. Select the best model among them.

Three variants of stepwise selection

- ▶ **Forward selection:** Starting from a *null model*, include variables one at a time, minimizing the RSS at each step.
- ▶ **Backward selection:** Starting from the *full model*, eliminate variables one at a time, choosing the one with the largest t-test p-value at each step.
- ▶ **Mixed selection:** Starting from a *null model*, include variables one at a time, minimizing the RSS at each step. If the p-value for some variable goes beyond a threshold, eliminate that variable.

Which subset of variables are important?

The output of a stepwise selection method is a range of models:

- ▶ {}
- ▶ {tv}
- ▶ {tv, newspaper}
- ▶ {tv, newspaper, radio}
- ▶ {tv, newspaper, radio, facebook}
- ▶ {tv, newspaper, radio, facebook, twitter}

6 choices are better than $2^6 = 64$. We use different *tuning methods* to decide which model to use; e.g. cross-validation, AIC, BIC.

AIC和BIC越小越好，直到AIC和BIC没有在变小为止
交叉验证本质就是一种对预测误差的较为精确的估计

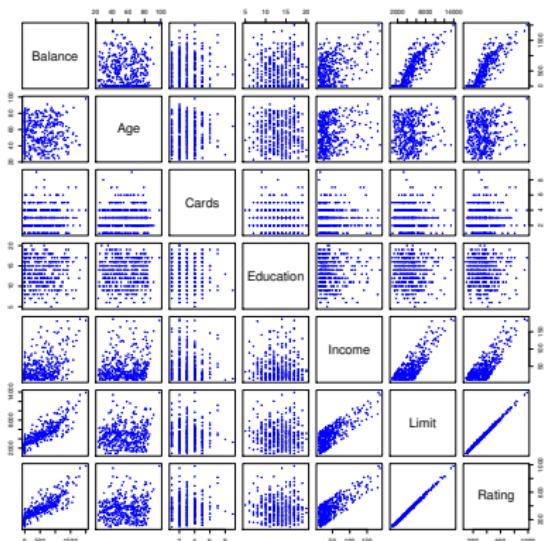
Which subset of variables are important?

When choosing a subset of the predictors, we have 2^p choices.

- ▶ **Forward selection:** Starting from a *null model*, include variables one at a time, minimizing the RSS at each step.
- ▶ **Backward selection:** Starting from the *full model*, eliminate variables one at a time, choosing the one with the largest p-value at each step.
- ▶ **Mixed selection:** Starting from a *null model*, include variables one at a time, minimizing the RSS at each step. If the p-value for some variable goes beyond a threshold, eliminate that variable.

Dealing with categorical or qualitative predictors

Example: Credit dataset



In addition, there are 4 qualitative variables:

- ▶ gender: male, female.
- ▶ student: student or not.
- ▶ status: married, single, divorced.
- ▶ ethnicity: African American, Asian, Caucasian.

Dealing with categorical or qualitative predictors

For each qualitative predictor, e.g. ethnicity:

- ▶ Choose a baseline category, e.g. African American
- ▶ For every other category, define a new predictor:
 - ▶ X_{Asian} is 1 if the person is Asian and 0 otherwise.
 - ▶ $X_{\text{Caucasian}}$ is 1 if the person is Caucasian and 0 otherwise.

The model will be:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_7 X_7 + \beta_{\text{Asian}} X_{\text{Asian}} + \beta_{\text{Caucasian}} X_{\text{Caucasian}} + \varepsilon.$$

β_{Asian} is the relative effect on balance for being Asian compared to the baseline category.

Dealing with categorical or qualitative predictors

- ▶ The model fit and predictions are independent of the choice of the baseline category.
- ▶ However, hypothesis tests derived from these variables are affected by the choice.
 - ▶ **Solution:** To check whether ethnicity is important, use an F -test for the hypothesis $\beta_{\text{Asian}} = \beta_{\text{Caucasian}} = 0$. This does not depend on the coding.
 - ▶ Other ways to encode qualitative predictors produce the same fit \hat{f} , but the coefficients have different interpretations.

How good are the predictions?

The function predict in R output predictions from a linear model;
eg. $x_0 = (5, 10, 15)$:

```
> predict(lm.fit, data.frame(lstat=c(5,10,15))),  
    interval="confidence")  
    fit      lwr      upr  
1 29.80  29.01  30.60  
2 25.05  24.47  25.63  
3 20.30  19.73  20.87
```

“Confidence intervals” reflect the uncertainty on $\hat{\beta}$; ie. confidence interval for $\hat{f}(x_0)$.

```
> predict(lm.fit, data.frame(lstat=c(5,10,15))),  
    interval="prediction")  
    fit      lwr      upr  
1 29.80  17.566  42.04  
2 25.05  12.828  37.28  
3 20.30  8.078  32.53
```

prediction明显比confidence更加宽
因为他还加上了irreducible的部分

“Prediction intervals” reflect uncertainty on $\hat{\beta}$ and the irreducible error ε as well; i.e. confidence interval for y_0 .

Recap

So far, we have:

- ▶ Defined Multiple Linear Regression
- ▶ Discussed how to test the importance of variables.
- ▶ Described one approach to choose a subset of variables.
- ▶ Explained how to code qualitative variables.
- ▶ Now, how do we evaluate model fit? Is the linear model any good? What can go wrong?

How good is the fit?

To assess the fit, we focus on the residuals.

R方反应的是预测值与真实值之间的线性相关系数

也可以是用
调整后的R方

- ▶ $R^2 = \text{Corr}^2(Y, \hat{Y})$, always increases as we add more variables.
- ▶ The residual standard error (RSE) does not always improve with more predictors:

$$\text{RSE} = \sqrt{\frac{1}{n - p - 1} \text{RSS}}.$$

- ▶ Visualizing the residuals can reveal phenomena that are not accounted for by the model.

Potential issues in linear regression

1. Interactions between predictors
2. Non-linear relationships
3. Correlation of error terms
4. Non-constant variance of error (heteroskedasticity).
5. Outliers
6. High leverage points
7. Colinearity

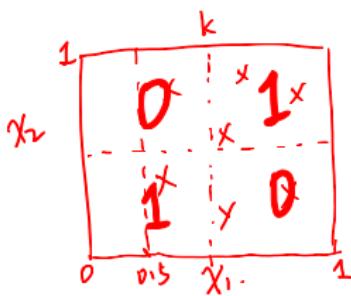
Interactions between predictors

Linear regression has an *additive* assumption:

$$\text{sales} = \beta_0 + \beta_1 \times \text{tv} + \beta_2 \times \text{radio} + \varepsilon$$

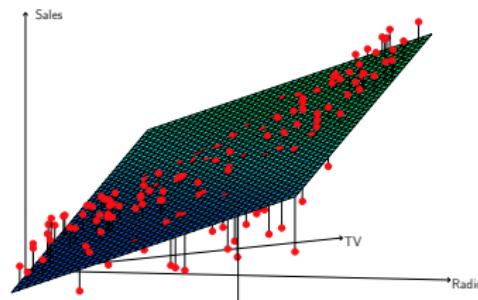
i.e. An increase of \$100 dollars in TV ads causes a fixed increase in sales, regardless of how much you spend on radio ads.

If we visualize the residuals, it is clear that this is false:



$$P(Y=1 | X_1=0.5) = 0.5$$

$$P(Y=2 | X_1=0.75) = 0.5$$



$$P(Y=1 | X_2=0.5) = 0.5$$

$$(X_1 - 0.5)(X_2 - 0.5) = X_1 X_2 - 0.5 X_2 - 0.5 X_1 + 0.5^2$$

Interactions between predictors

One way to deal with this is to include multiplicative variables in the model:

$$\text{sales} = \beta_0 + \beta_1 \times \text{tv} + \beta_2 \times \text{radio} + \beta_3 \times (\text{tv} \cdot \text{radio}) + \varepsilon$$

The **interaction variable** is high when both tv and radio are high.

Interactions between predictors

R makes it easy to include interaction variables in the model:

```
> lm.fit=lm(Sales~.+Income:Advertising+Price:Age,data=Carseats)
> summary(lm.fit)

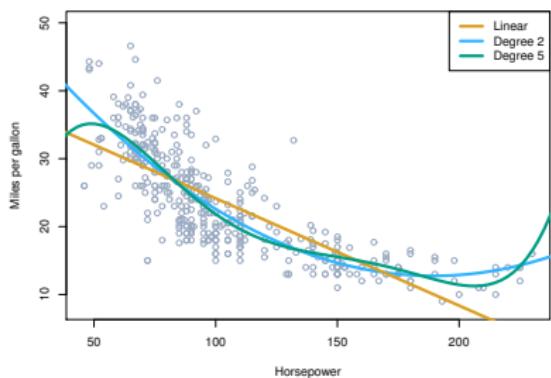
Call:
lm(formula = Sales ~ . + Income:Advertising + Price:Age, data =
Carseats)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.921 -0.750  0.018  0.675  3.341 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 6.575565  1.008747   6.52  2.2e-10 ***
CompPrice    0.092937  0.004118  22.57 < 2e-16 ***
Income       0.010894  0.002604   4.18  3.6e-05 ***
Advertising  0.070246  0.022609   3.11  0.00203 ** 
Population   0.000159  0.000368   0.43  0.66533  
Price        -0.100806 0.007440  -13.55 < 2e-16 ***
ShelveLocGood 4.848676  0.152838   31.72 < 2e-16 ***
ShelveLocMedium 1.953262  0.125768   15.53 < 2e-16 ***
Age          -0.057947  0.015951  -3.63  0.00032 *** 
Education    -0.020852  0.019613  -1.06  0.28836  
UrbanYes     0.140160  0.112402   1.25  0.21317  
USYes        -0.157557  0.148923  -1.06  0.29073  
Income:Advertising 0.000751  0.000278   2.70  0.00729 ** 
Price:Age     0.000107  0.000133   0.80  0.42381  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Non-linearities

Example: Auto dataset.



A scatterplot between a predictor and the response may reveal a non-linear relationship.

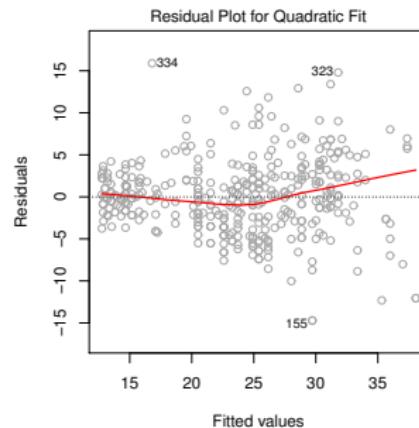
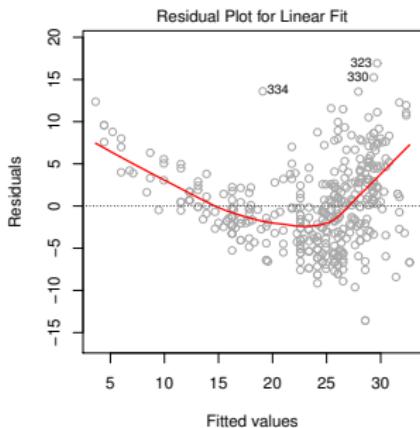
Solution: include polynomial terms in the model.

$$\begin{aligned} \text{MPG} = & \beta_0 + \beta_1 \times \text{horsepower} + \varepsilon \\ & + \beta_2 \times \text{horsepower}^2 + \varepsilon \\ & + \beta_3 \times \text{horsepower}^3 + \varepsilon \\ & + \dots + \varepsilon \end{aligned}$$

Non-linearities

In 2 or 3 dimensions, this is easy to visualize. What do we do when we have too many predictors?

Plot the residuals against the *response* and look for a pattern:



Correlation of error terms

自相关

1. 自相关不影响OLS估计量的线性和无偏性，但使之失去有效性，当数据不断增加时，越来越接近真实数据
2. 自相关的系数估计量将有相当大的方差
3. 自相关系数的T检验不显著
4. 模型的预测功能失效

We assumed that the errors for each sample are independent:

$$y_i = f(x_i) + \varepsilon_i \quad ; \quad \varepsilon_i \sim \mathcal{N}(0, \sigma) \text{ i.i.d.}$$

What if this breaks down?

The main effect is that this invalidates any assertions about
Standard Errors, confidence intervals, and hypothesis tests:

Example: Suppose that by accident, we double the data (we use each sample twice). Then, the standard errors would be artificially smaller by a factor of $\sqrt{2}$.

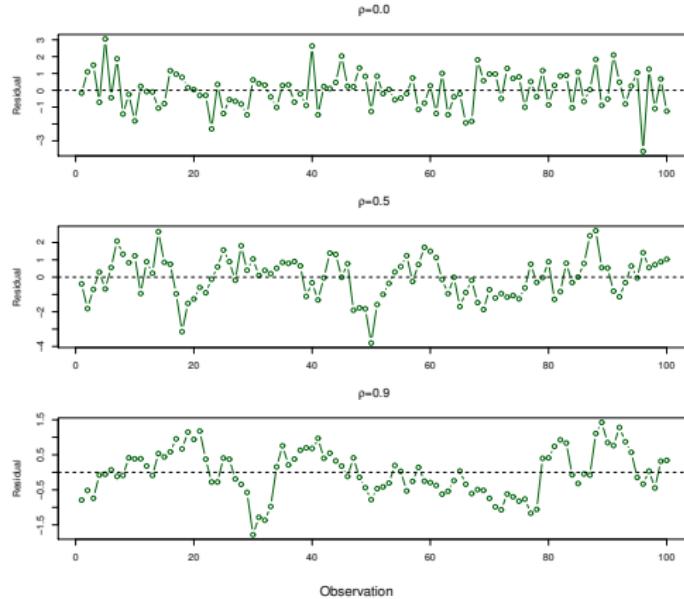
Correlation of error terms

When could this happen in real life:

- ▶ **Time series:** Each sample corresponds to a different point in time. The errors for samples that are close in time are correlated.
- ▶ **Spatial data:** Each sample corresponds to a different location in space.
- ▶ Study on predicting height from weight at birth. Suppose some of the subjects in the study are in the same family, their shared environment could make them deviate from $f(x)$ in similar ways.

Correlation of error terms

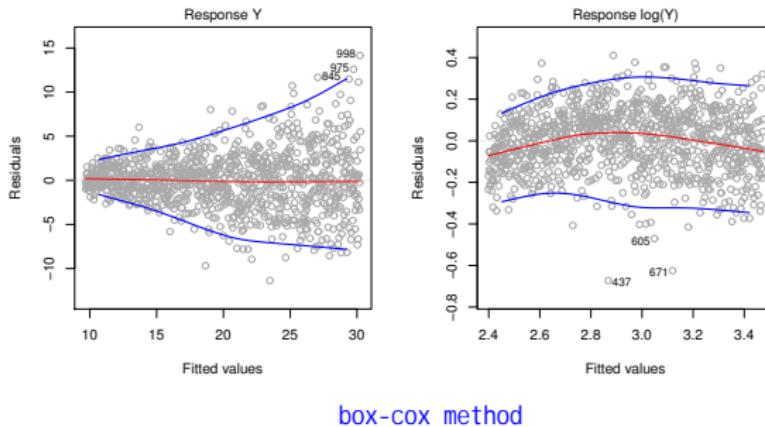
Simulations of time series with increasing correlations between ε_i .



Non-constant variance of error (heteroskedasticity)

The variance of the error depends on the input.

To diagnose this, we can plot residuals vs. fitted values:

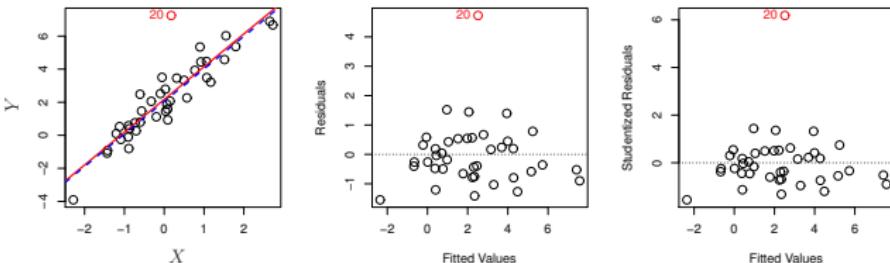


box-cox method

Solution: If the trend in variance is relatively simple, we can transform the response using a **logarithm**, for example.

Outliers

Outliers are points with very high errors.



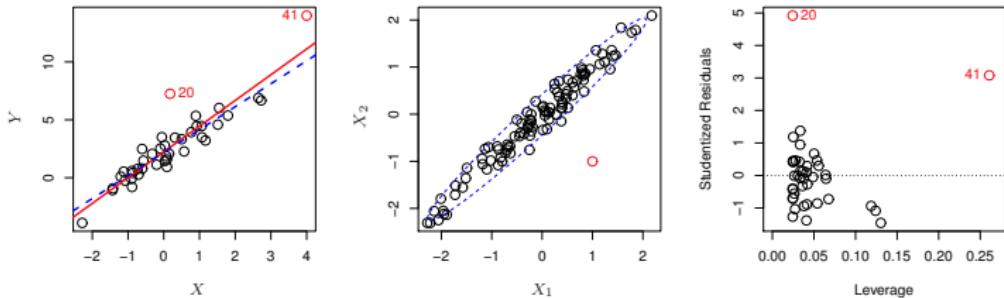
While they may not affect the fit, they might affect our assessment of model quality.

Possible solutions:

- ▶ If we believe an outlier is due to an error in data collection, we can remove it.
- ▶ An outlier might be evidence of a missing predictor, or the need to specify a more complex model.

High leverage points

Some samples with extreme inputs have an outsized effect on $\hat{\beta}$.

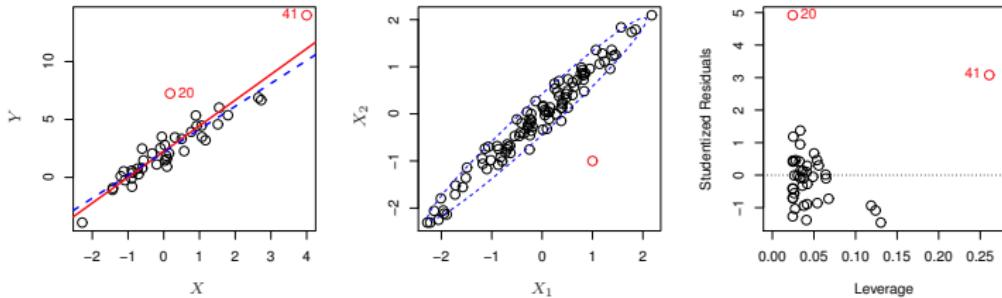


This can be measured with the **leverage statistic** or **self influence**:

$$h_{ii} = \frac{\partial \hat{y}_i}{\partial y_i} = (\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T)_{i,i} \in [1/n, 1].$$

High leverage points

Some samples with extreme inputs have an outsized effect on $\hat{\beta}$.

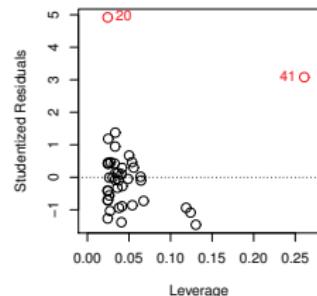
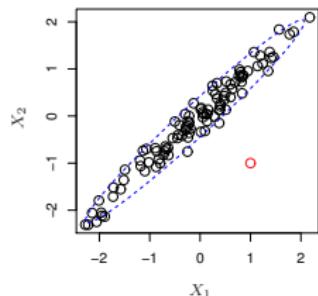
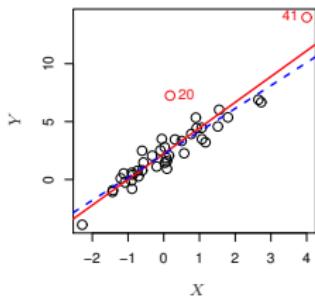


This can be measured with the **leverage statistic** or **self influence**:

$$h_{ii} = \frac{\partial \hat{y}_i}{\partial y_i} = (\underbrace{\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T}_{\text{Hat matrix}})_{i,i} \in [1/n, 1].$$

Studentized residuals

- ▶ The residual $\hat{\epsilon}_i = y_i - \hat{y}_i$ is an estimate for the noise ϵ_i .
- ▶ The standard error of $\hat{\epsilon}_i$ is $\sigma\sqrt{1 - h_{ii}}$.
- ▶ A **studentized residual** is $\hat{\epsilon}_i$ divided by its standard error.
- ▶ It follows a Student-t distribution with $n - p - 2$ degrees of freedom.

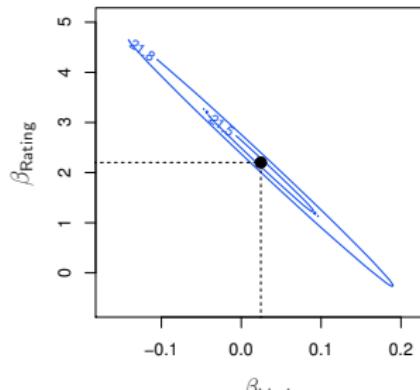
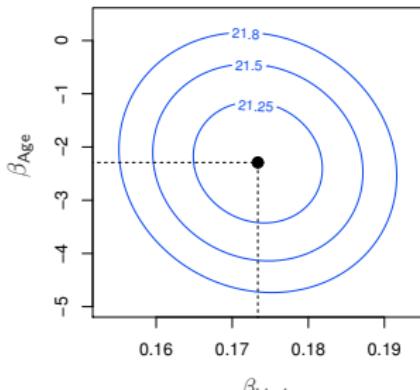


Collinearity

Problem: The coefficients become *unidentifiable*. Consider the extreme case of using two identical predictors limit:

$$\begin{aligned}\text{balance} &= \beta_0 + \beta_1 \times \text{limit} + \beta_2 \times \text{limit} \\ &= \beta_0 + (\beta_1 + 100) \times \text{limit} + (\beta_2 - 100) \times \text{limit}\end{aligned}$$

The fit $(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$ is just as good as $(\hat{\beta}_0, \hat{\beta}_1 + 100, \hat{\beta}_2 - 100)$.



Collinearity

If 2 variables are collinear, we can easily diagnose this using their correlation.

A group of q variables is **multilinear** if these variables "contain less information" than q independent variables. Pairwise correlations may not reveal multilinear variables.

The Variance Inflation Factor (VIF) measures how *necessary* a variable is, or how predictable it is given the other variables:

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R_{X_j|X_{-j}}^2},$$

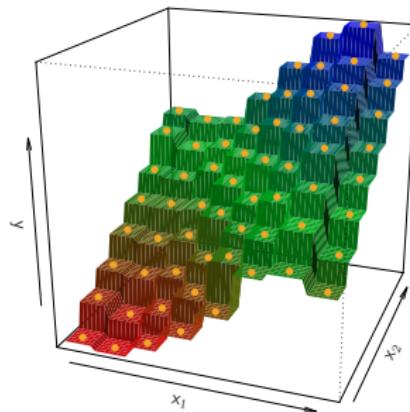
where $R_{X_j|X_{-j}}^2$ is the R^2 statistic for Multiple Linear regression of the predictor X_j onto the remaining predictors.

Comparing Linear Regression to K -nearest neighbors

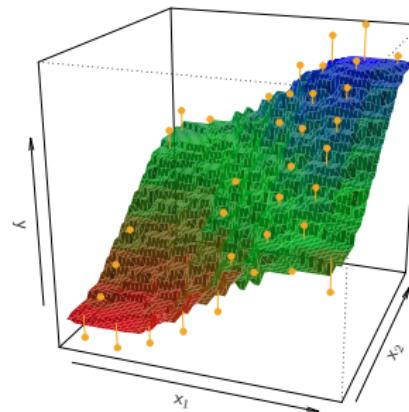
Linear regression: prototypical parametric method.

KNN regression: prototypical nonparametric method.

$$\hat{f}(x) = \frac{1}{K} \sum_{i \in N_K(x)} y_i$$



$$K = 1$$



$$K = 9$$

Comparing Linear Regression to K -nearest neighbors

Linear regression: prototypical parametric method.

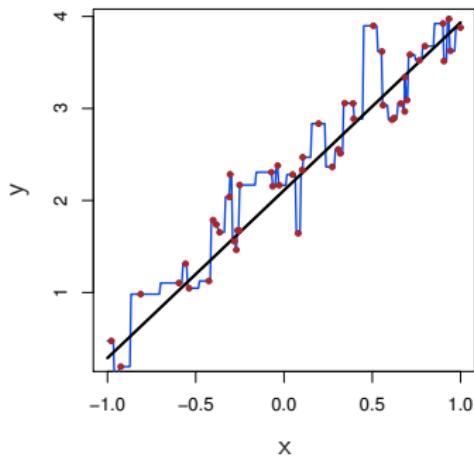
KNN regression: prototypical nonparametric method.

Long story short:

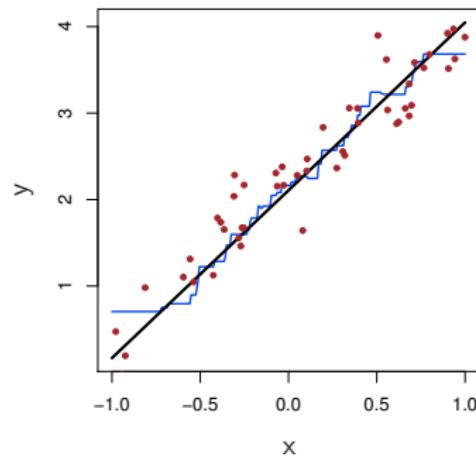
- ▶ KNN is only better when the function f is not linear.
- ▶ When n is not much larger than p , even if f is nonlinear, Linear Regression can outperform KNN. KNN has smaller bias, but this comes at a price of higher variance.

KNN estimates for a simulation from a linear model

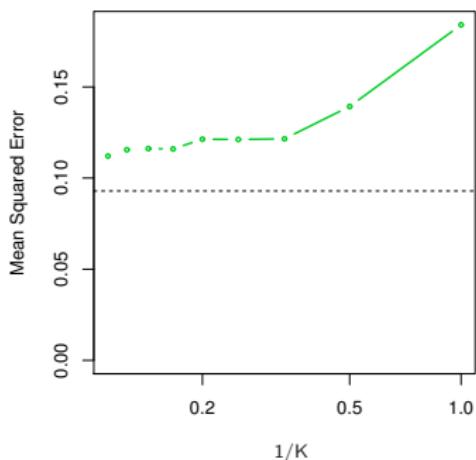
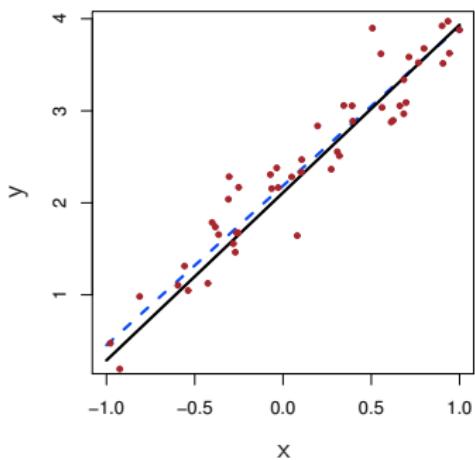
$$K = 1$$



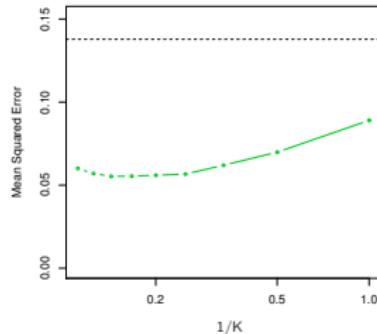
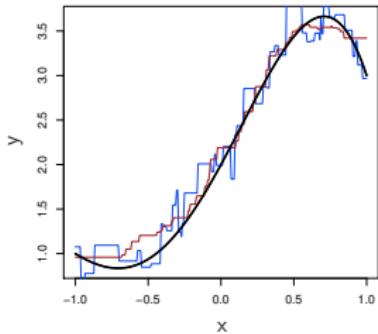
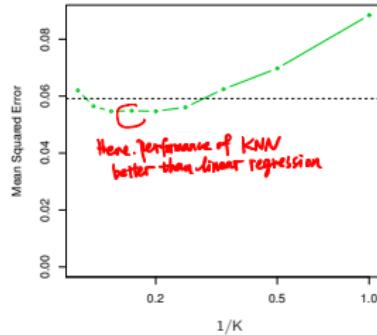
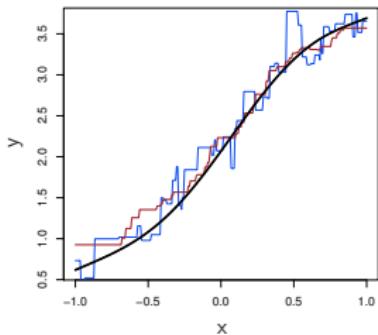
$$K = 9$$



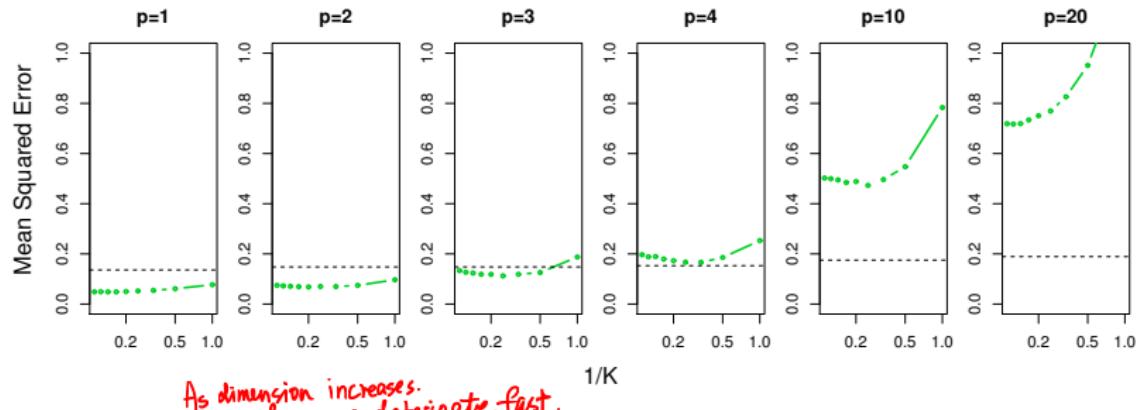
Linear models dominate KNN



Increasing deviations from linearity



When there are more predictors than observations, Linear Regression dominates



When $p \gg n$, each sample has no nearest neighbors, this is known as the *curse of dimensionality*. The variance of KNN regression is very large.

Next time: Classification

Supervised learning with a **qualitative or categorical** response.

Just as common, if not more common than regression:

- ▶ *Medical diagnosis*: Given the symptoms a patient shows, predict which of 3 conditions they are attributed to.
- ▶ *Online banking*: Determine whether a transaction is fraudulent or not, on the basis of the IP address, client's history, etc.
- ▶ *Web searching*: Based on a user's history, location, and the string of a web search, predict which link a person is likely to click.
- ▶ *Online advertising*: Predict whether a user will click on an ad or not.

Thanks to Sergio Bacallado and Peter Orbanz
for sharing the slides.

Lecture 6: Classification

Reading: Sections 4.3, 4.4

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
February 2, 2018

Classification problems

Supervised learning with a **qualitative or categorical** response.

Just as common, if not more common than regression:

- ▶ *Medical diagnosis*: Given the symptoms a patient shows, predict which of 3 conditions they are attributed to.
- ▶ *Online banking*: Determine whether a transaction is fraudulent or not, on the basis of the IP address, client's history, etc.
- ▶ *Web searching*: Based on a user's history, location, and the string of a web search, predict which link a person is likely to click.
- ▶ *Online advertising*: Predict whether a user will click on an ad or not.

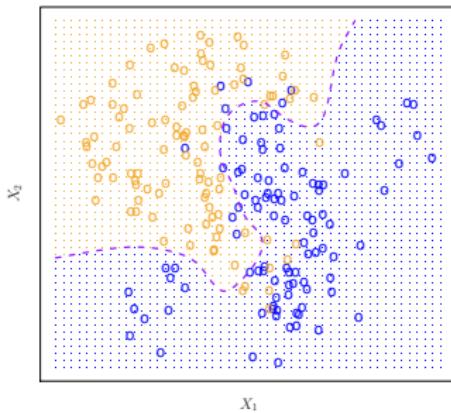
$$(X, Y) \quad P(Y=1 | X=x) + P(Y=0 | X=x) = 1$$

$\uparrow Y \in \{0, 1\}$

Prediction: based on input (background)
know which class does this belong to

$P(Y=1 | X=x) = \frac{P(X=x | Y=1) P(Y=1)}{P(X=x)}$

Classification problem



ISL Figure 2.13

Recall:

- ▶ $X = (X_1, X_2)$ are inputs.
- ▶ Color $Y \in \{\text{Yellow}, \text{Blue}\}$ is the output.
- ▶ (X, Y) have a joint distribution.
- ▶ Purple line is *Bayes boundary* — the best we could do if we knew the joint distribution of (X, Y)

Review: Bayes classifier

Suppose $P(Y | X)$ is known. Then, given an input x_0 , we predict the response

最大化他的后验预测分布

$$\hat{y}_0 = \operatorname{argmax}_y P(Y = y | X = x_0).$$

The Bayes classifier minimizes the expected 0-1 loss:

$$E \left[\frac{1}{m} \sum_{i=1}^m \mathbf{1}(\hat{y}_i \neq y_i) \right]$$

这是一种expected loss

This minimum 0-1 loss (the best we can hope for) is the **Bayes error rate**.

Example: Spam Filtering

Representing emails

$$P(X_1, \dots, X_n | Y=1) = P(X_1|Y=1) \cdot P(X_2|Y=1) \cdot \dots \cdot P(X_n|Y=1)$$

a collection of words

Assumption:
 x_1, \dots, x_n are independent.

- $Y = \{ \text{spam, email} \}$
- $X = \mathbb{R}^d$
- Each axis is labelled by one possible word.
- $d = \text{number of distinct words in vocabulary}$
- $x_j = \text{number of occurrences of word } j \text{ in email represented by } x$

For example, if axis j represents the term "the", $x_j = 3$ means that "the" occurs three times in the email x . This representation is called a **vector space model of text**.

Example dimensions

	george	you	your	hp	free	hpl	!	our	re	edu
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29

With Bayes equation

关键在于给定y的情况下x的条件概率

$$f(x) = \operatorname{argmax}_{y \in \{\text{spam, email}\}} P(y|x) = \operatorname{argmax}_{y \in \{\text{spam, email}\}} p(x|y)P(y)$$

Naive Bayes

Simplifying assumption

The classifier is called a **naive Bayes classifier** if it assumes

$$p(\mathbf{x}|y) = \prod_{j=1}^d p_j(x_i|y) ,$$

i.e. if it treats the individual dimensions of \mathbf{x} as conditionally **independent** given y .

In spam example

- ▶ Corresponds to the assumption that the number of occurrences of a word carries information about y .
- ▶ **Co-occurrences (how often do given combinations of words occur?) is neglected.** 因为视各个变量在给定 y 之后的条件分布是独立的。所以不能考察它们之间的交互作用。

Estimation

Class prior

The distribution $P(y)$ is easy to estimate from training data:

$$P(y) = \frac{\text{\#observations in class } y}{\text{\#observations}}$$

Class-conditional distributions

The class conditionals $p(x|y)$ usually require a modeling assumption. Under a given model:

- ▶ Separate the training data into classes.
- ▶ Estimate $p(x|y)$ on class y by maximum likelihood.

Strategy: estimate $P(Y | X)$

If we have a good estimate for the conditional probability $\hat{P}(Y | X)$, we can use the classifier:

$$\hat{y}_0 = \operatorname{argmax}_y \hat{P}(Y = y | X = x_0).$$

Suppose Y is a binary variable. Could we use a linear model?

$$P(Y = 1 | X) = \beta_0 + \beta_1 \underline{X_1} + \cdots + \beta_1 \underline{X_p}$$

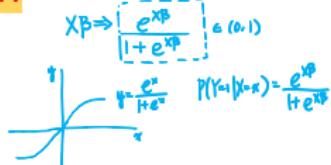
They can be any number.

can not guarantee $0 \leq P \leq 1$

Problems:

- ▶ This would allow probabilities < 0 and > 1 .
- ▶ Difficult to extend to more than 2 categories.

Logistic regression



We model the joint probability as:

$$P(Y = 1 | X) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}},$$

$$P(Y = 0 | X) = \frac{1}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}.$$

This is the same as using a linear model for the log odds:

$$\log \left[\frac{P(Y = 1 | X)}{P(Y = 0 | X)} \right] = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p.$$

Fitting logistic regression

The training data is a list of pairs $(y_1, x_1), (y_2, x_2), \dots, (y_n, x_n)$. In the linear model

$$\log \left[\frac{P(Y = 1 | X)}{P(Y = 0 | X)} \right] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p,$$

we don't observe the left hand side.

We cannot use a least squares fit.

Fitting logistic regression

Solution:

The likelihood is the probability of the training data, for a fixed set of coefficients β_0, \dots, β_p :

$$\prod_{i=1}^n P(Y = y_i | X = x_i) \\ = \underbrace{\prod_{i:y_i=1} \frac{e^{\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}}} \underbrace{\text{Probability of responses } = 1}_{\text{Probability of responses } = 0}}$$

- ▶ Choose estimates $\hat{\beta}_0, \dots, \hat{\beta}_p$ which maximize the likelihood.
- ▶ Solved with numerical methods (e.g. Newton's algorithm).

Logistic regression in R

```
> glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume ,  
    data=Smarket,family=binomial)  
> summary(glm.fit)
```

Call:

```
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5  
+ Volume, family = binomial, data = Smarket)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.45	-1.20	1.07	1.15	1.33

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.12600	0.24074	-0.52	0.60
Lag1	-0.07307	0.05017	-1.46	0.15
Lag2	-0.04230	0.05009	-0.84	0.40
Lag3	0.01109	0.04994	0.22	0.82
Lag4	0.00936	0.04997	0.19	0.85
Lag5	0.01031	0.04951	0.21	0.83
Volume	0.13544	0.15836	0.86	0.39

Logistic regression in R

- ▶ We can estimate the Standard Error of each coefficient.
- ▶ The z -statistic is the equivalent of the t -statistic in linear regression:

$$z = \frac{\hat{\beta}_j}{\text{SE}(\hat{\beta}_j)}.$$

*only when n very large
can say sth. about the distribution*

- ▶ The p -values are test of the null hypothesis $\beta_j = 0$ (Wald's test).
- ▶ Other possible hypothesis tests: likelihood ratio test (chi-square distribution).

Example: Predicting credit card default

Predictors:

- ▶ student: 1 if student, 0 otherwise. (*categorical*)
- ▶ balance: credit card balance.
- ▶ income: person's income.

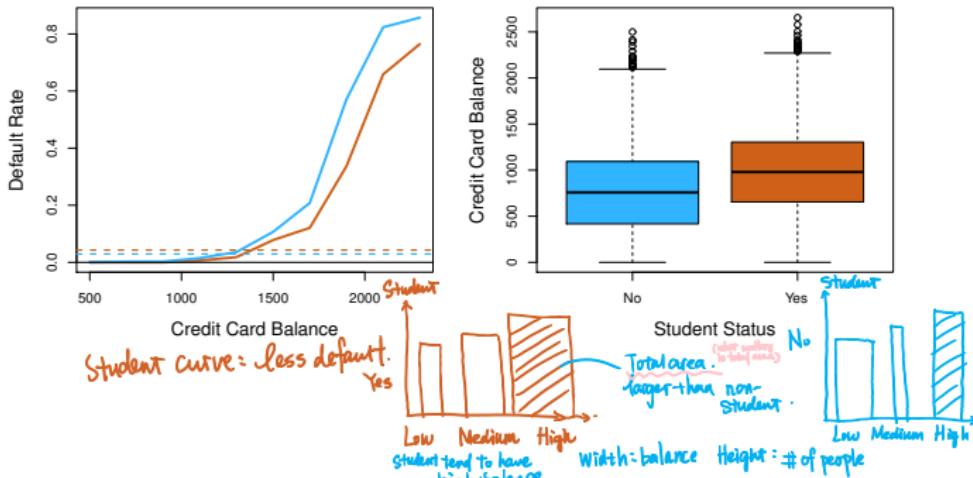
In this dataset, there is *confounding*, but little collinearity.

- ▶ Students tend to have higher balances. So, balance is explained by student, but not very well.
- ▶ People with a high balance are more likely to default.
- ▶ Among people with a given balance, students are less likely to default.

Example: Predicting credit card default

Predictors:

- ▶ student: 1 if student, 0 otherwise.
- ▶ balance: credit card balance.
- ▶ income: person's income.



Example: Predicting credit card default

Logistic regression using only balance:

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

Logistic regression using only student:

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	<0.0001
student [Yes]	0.4049	0.1150	3.52	0.0004 *

student more likely to default

Logistic regression using all 3 predictors:

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001 *
income	0.0030	0.0082	0.37	0.7115
student [Yes]	-0.6468	0.2362	-2.74	0.0062

If use all the variables.
for student will get opposite conclusion: student less likely to default.

Extending logistic regression to more than 2 categories

Multinomial logistic regression:

Suppose Y takes values in $\{1, 2, \dots, K\}$, then we use a linear model for the log odds against a **baseline category** (e.g. 1):

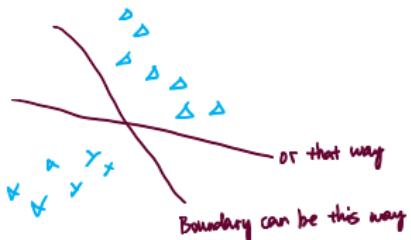
$$\log \left[\frac{P(Y = 2 | X)}{P(Y = 1 | X)} \right] = \beta_{0,2} + \beta_{1,2}X_1 + \cdots + \beta_{p,2}X_p,$$

.....

$$\log \left[\frac{P(Y = K | X)}{P(Y = 1 | X)} \right] = \beta_{0,K} + \beta_{1,K}X_1 + \cdots + \beta_{p,K}X_p.$$

Some issues with logistic regression

- ▶ The coefficients become unstable when there is collinearity. Furthermore, this affects the convergence of the fitting algorithm.
- ▶ When the classes are well separated, the coefficients become unstable. This is always the case when $p \geq n - 1$.



Main strategy in Chapter 4

Find an estimate $\hat{P}(Y | X)$. Then, given an input x_0 , we predict the response as in a Bayes classifier:

$$\hat{y}_0 = \operatorname{argmax}_y \hat{P}(Y = y | X = x_0).$$

Linear Discriminant Analysis (LDA)

Strategy: Instead of estimating $P(Y | X)$, we will estimate:

1. $\hat{P}(X | Y)$: Given the response, what is the distribution of the inputs.
2. $\hat{P}(Y)$: How likely are each of the categories.

Then, we use **Bayes rule** to obtain the estimate:

$$\hat{P}(Y = k | X = x) = \frac{\hat{P}(X = x | Y = k)\hat{P}(Y = k)}{\hat{P}(X = x)}$$

likelihood matters.
determine which dist.
to u

↑
doesn't matter.

Linear Discriminant Analysis (LDA)

Strategy: Instead of estimating $P(Y | X)$, we will estimate:

1. $\hat{P}(X | Y)$: Given the response, what is the distribution of the inputs.
2. $\hat{P}(Y)$: How likely are each of the categories.

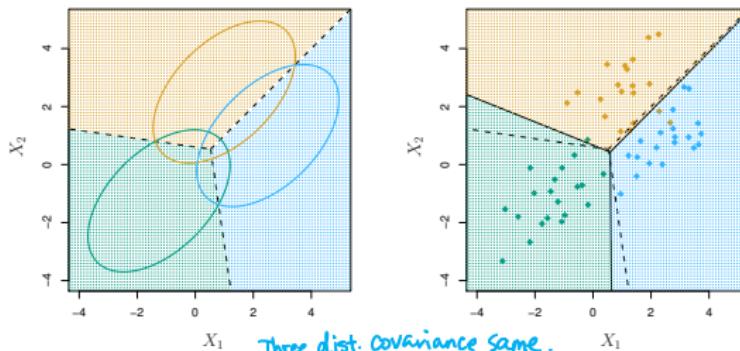
Then, we use *Bayes rule* to obtain the estimate:

$$\hat{P}(Y = k | X = x) = \frac{\hat{P}(X = x | Y = k)\hat{P}(Y = k)}{\sum_j \hat{P}(X = x | Y = j)\hat{P}(Y = j)}$$

Linear Discriminant Analysis (LDA)

Strategy: Instead of estimating $P(Y | X)$, we will estimate:

1. We model $\hat{P}(X = x | Y = k) = \hat{f}_k(x)$ as a **Multivariate Normal Distribution**: with the same variance



2. $\hat{P}(Y = k) = \hat{\pi}_k$ is estimated by the fraction of training samples of class k .

LDA has linear decision boundaries

Suppose that:

- ▶ We know $P(Y = k) = \pi_k$ exactly.
- ▶ $P(X = x|Y = k)$ is Multivariate Normal with density:

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1} (x-\mu_k)}$$

μ_k : Mean of the inputs for category k .

Σ : Covariance matrix (**common to all categories**).

Then, what is the Bayes classifier?

LDA has linear decision boundaries

By Bayes rule, the probability of category k , given the input x is:

$$P(Y = k \mid X = x) = \frac{f_k(x)\pi_k}{P(X = x)}$$

The denominator does not depend on the response k , so we can write it as a constant:

$$P(Y = k \mid X = x) = C \times f_k(x)\pi_k$$

Now, expanding $f_k(x)$:

$$P(Y = k \mid X = x) = \frac{C\pi_k}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

LDA has linear decision boundaries

$$P(Y = k \mid X = x) = \frac{C\pi_k}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

Now, let us absorb everything that does not depend on k into a constant C' :

$$P(Y = k \mid X = x) = C'\pi_k e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

and take the logarithm of both sides:

$$\log P(Y = k \mid X = x) = \log C' + \log \pi_k - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k).$$

This is the same for every category, k .

LDA has linear decision boundaries

$$P(Y = k \mid X = x) = \frac{C\pi_k}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

Now, let us absorb everything that does not depend on k into a constant C' :

$$P(Y = k \mid X = x) = C' \pi_k e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

and take the logarithm of both sides:

$$\log P(Y = k \mid X = x) = \underbrace{\log C' + \log \pi_k}_{\text{maximize this blue part.}} - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k).$$

This is the same for every category, k .

So we want to find the maximum of this over k .

LDA has linear decision boundaries

Goal, maximize the following over k :

$$\begin{aligned} & \log \pi_k - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k). \\ &= \log \pi_k - \frac{1}{2} [x^T \Sigma^{-1} x + \mu_k^T \Sigma^{-1} \mu_k] + x^T \Sigma^{-1} \mu_k \\ &= C'' + \log \pi_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + x^T \Sigma^{-1} \mu_k \end{aligned}$$

We define the objective:

$$\delta_k(x) = \log \pi_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + x^T \Sigma^{-1} \mu_k$$

At an input x , we predict the response with the highest $\delta_k(x)$.

LDA has linear decision boundaries

What is the decision boundary? It is the set of points in which 2 classes do just as well:

$$\delta_k(x) = \delta_\ell(x)$$

$$\log \pi_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + x^T \Sigma^{-1} \mu_k = \log \pi_\ell - \frac{1}{2} \mu_\ell^T \Sigma^{-1} \mu_\ell + x^T \Sigma^{-1} \mu_\ell$$

if x is fixed *Want to max:*

$$\begin{aligned} & \log \pi_k - \frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \\ &= \log \pi_k - \frac{1}{2} (x^T \Sigma^{-1} x - 2\mu_k^T \Sigma^{-1} x + \mu_k^T \Sigma^{-1} \mu_k) \\ &= C + \underbrace{\log \pi_k + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k}_{S(x)} \\ &= \log \pi_k + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \\ &= \log \pi_k + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \\ &\quad \text{linear in } x + \text{(decision boundary is linear)} \end{aligned}$$

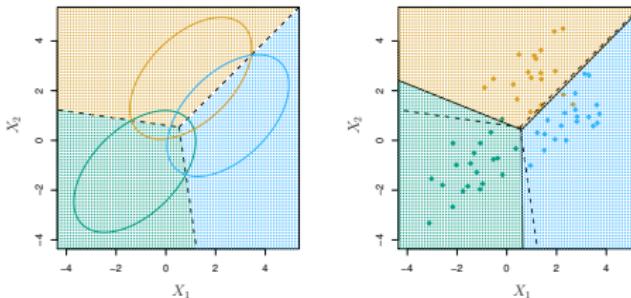
LDA has linear decision boundaries

What is the decision boundary? It is the set of points in which 2 classes do just as well:

$$\delta_k(x) = \delta_\ell(x)$$

$$\log \pi_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \textcolor{red}{x}^T \Sigma^{-1} \mu_k = \log \pi_\ell - \frac{1}{2} \mu_\ell^T \Sigma^{-1} \mu_\ell + \textcolor{red}{x}^T \Sigma^{-1} \mu_\ell$$

This is a linear equation in $\textcolor{red}{x}$.



Estimating π_k

$$\hat{\pi}_k = \frac{\#\{i ; y_i = k\}}{n}$$

In English, the fraction of training samples of class k .

$\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$.

$$\frac{\hat{\mu}_k}{n-k} \sum_i (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

Estimating the parameters of $f_k(x)$

Estimate the center of each class μ_k :

$$\hat{\mu}_k = \frac{1}{\#\{i ; y_i = k\}} \sum_{i ; y_i = k} x_i$$

Estimate the common covariance matrix Σ :

- One predictor ($p = 1$):

所有的数据都放进来一起计算

$$\hat{\sigma}^2 = \frac{1}{n - K} \sum_{k=1}^K \sum_{i ; y_i = k} (x_i - \hat{\mu}_k)^2.$$

- Many predictors ($p > 1$): Compute the vectors of deviations $(x_1 - \hat{\mu}_{y_1}), (x_2 - \hat{\mu}_{y_2}), \dots, (x_n - \hat{\mu}_{y_n})$ and use an unbiased estimate of its covariance matrix, Σ .

$\text{sigma}^2 = 1/(n-K) \sum^K_{k=1} t(X_k - U_k) * (X_k - U_k)$

LDA prediction

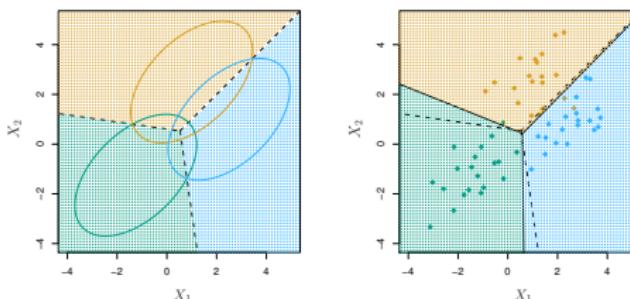
For an input x , predict the class with the largest:

$$\hat{\delta}_k(x) = \log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + x^T \hat{\Sigma}^{-1} \hat{\mu}_k$$

The decision boundaries are defined by:

$$\log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + x^T \hat{\Sigma}^{-1} \hat{\mu}_k = \log \hat{\pi}_\ell - \frac{1}{2} \hat{\mu}_\ell^T \hat{\Sigma}^{-1} \hat{\mu}_\ell + x^T \hat{\Sigma}^{-1} \hat{\mu}_\ell$$

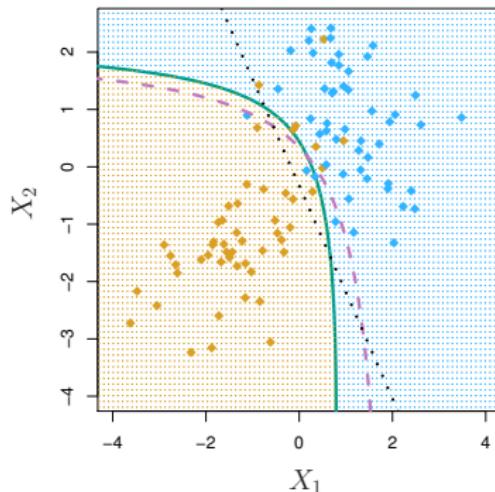
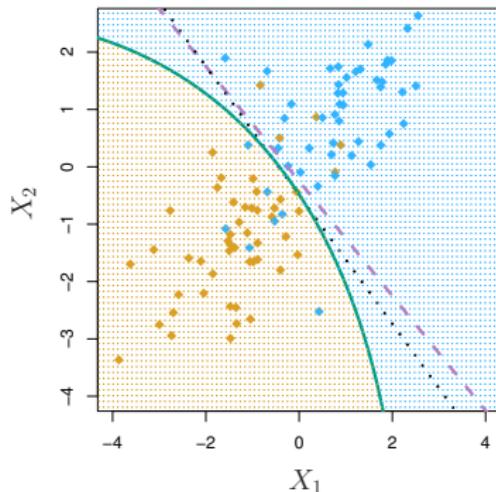
Solid lines in:



Quadratic discriminant analysis (QDA)

The assumption that the inputs of every class have the same covariance Σ can be quite restrictive:

Purple: decision boundary



Quadratic discriminant analysis (QDA)

In quadratic discriminant analysis we estimate a mean $\hat{\mu}_k$ and a covariance matrix $\hat{\Sigma}_k$ for each class separately.

Given an input, it is easy to derive an objective function:

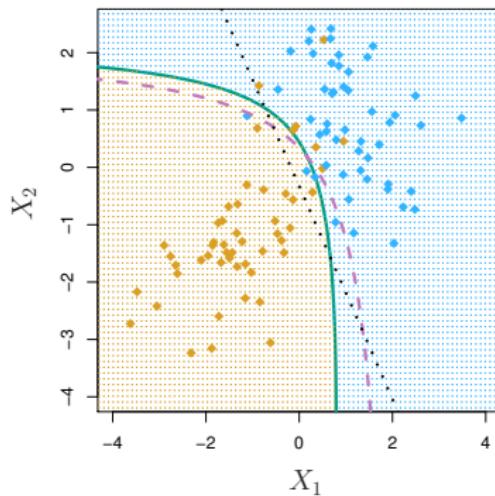
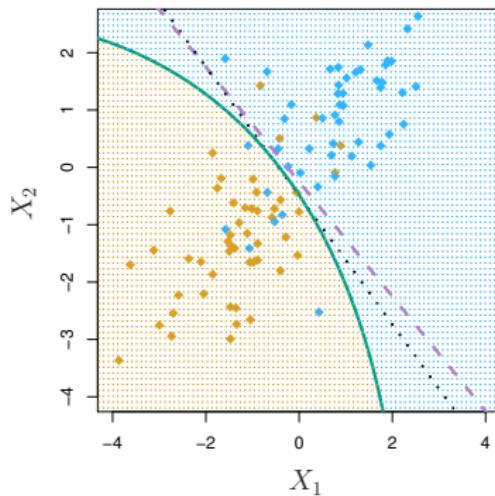
$$\delta_k(x) = \log \pi_k - \frac{1}{2} \mu_k^T \Sigma_k^{-1} \mu_k + \underbrace{x^T \Sigma_k^{-1} \mu_k}_{\text{Add this two parts into } \delta(x) \text{ [LDA].}} - \frac{1}{2} x^T \Sigma_k^{-1} x - \underbrace{\frac{1}{2} \log |\Sigma_k|}$$

This objective is now quadratic in x and so are the decision boundaries.

$$\sigma^2_K = \sum_{i:y_i=k} t(X_k - U_k)^* (X_k - U_k)$$

Quadratic discriminant analysis (QDA)

- ▶ Bayes boundary (---)
- ▶ LDA (· · · · ·)
- ▶ QDA (—).



Evaluating a classification method

We have talked about the 0-1 loss:

$$\frac{1}{m} \sum_{i=1}^m \mathbf{1}(y_i \neq \hat{y}_i).$$

It is possible to make the wrong prediction for some classes more often than others. The 0-1 loss doesn't tell you anything about this.

A much more informative summary of the error is a **confusion matrix**:

		<i>Predicted class</i>		Total
<i>True class</i>	– or Null	+ or Non-null		
	– or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
Total		N*	P*	

Example. Predicting default

Used LDA to predict credit card default in a dataset of 10K people.

Predicted "yes" if $P(\text{default} = \text{yes} | X) > 0.5$.

实操中这个是可以被实现的。运用LDA，可以计算出 $p(y|x)$ ，那么只要关注 $P(y=1|x)$ 是可以计算出具体的概率的。

want to avoid
False Negatives
(predict = -
True = +)

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
Total		9,667	333	10,000

negative or positive 是值判断是正还是负

True or False 是说这样的判断是不是对的

FN: 真的被判断为假的；FP就是：假的判断为真

- The error rate among people who do **not** default (**false positive rate**) is very low. FP: 放在题目中就是本来是会有违约风险的人，被判定成不会违约 ==> 坏账率高
FN: 放在题目中就是本来没有违约风险的人，被判定成会有违约风险 ==> 业务流式
- However, the rate of **false negatives** is 76%. #fn/#n
- It is possible that false negatives are a bigger source of concern!
- One possible solution: Change the **threshold**.

Example. Predicting default

Changing the threshold to 0.2 makes it easier to classify to “yes”.

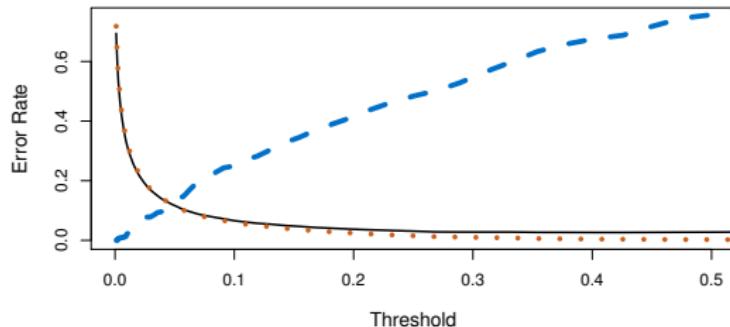
Predicted “yes” if $P(\text{default} = \text{yes}|X) > 0.2$.

		True default status		Total
		No	Yes	
Predicted default status	No	9,432	138	9,570
	Yes	235	195	430
Total	9,667	333	10,000	

Note that the rate of false positives became higher! That is the price to pay for fewer false negatives.

Example. Predicting default

Let's visualize the dependence of the error on the threshold:



$$FNR = FN / (FN + TP)$$

$$FPR = FP / (FP + TN)$$

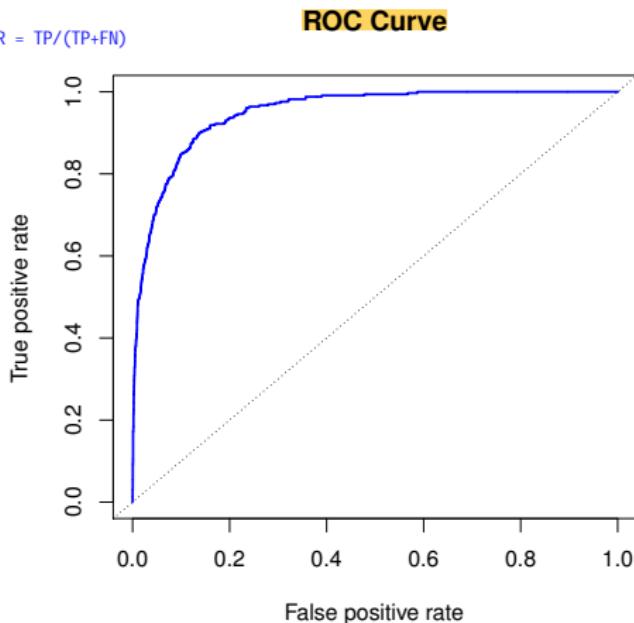
$$\text{total error rate} = (TN + TP) / (TP + TN + FP + FN)$$

首先：FNR与FPR 此消彼长是一定的。
其次：total error rate和谁一起涨是不一定

- ▶ —— False negative rate (error for defaulting customers)
- ▶ False positive rate (error for non-defaulting customers)
- ▶ — 0-1 loss or total error rate.

Example. The ROC curve

TPR = $TP/(TP+FN)$



- ▶ Displays the performance of the method for any choice of threshold.
- ▶ The area under the curve (AUC) measures the quality of the classifier:
 - ▶ 0.5 is the AUC for a random classifier
 - ▶ The closer AUC is to 1, the better.

Thinking about the loss function is important

Most of the **regression** methods we've studied aim to minimize the RSS, while **classification** methods aim to minimize the 0-1 loss.

In classification, we often care about certain kinds of error more than others; i.e. the natural loss function is not the 0-1 loss.

Even if we use a method which minimizes a certain kind of training error, we can *tune* it to optimize our true loss function.

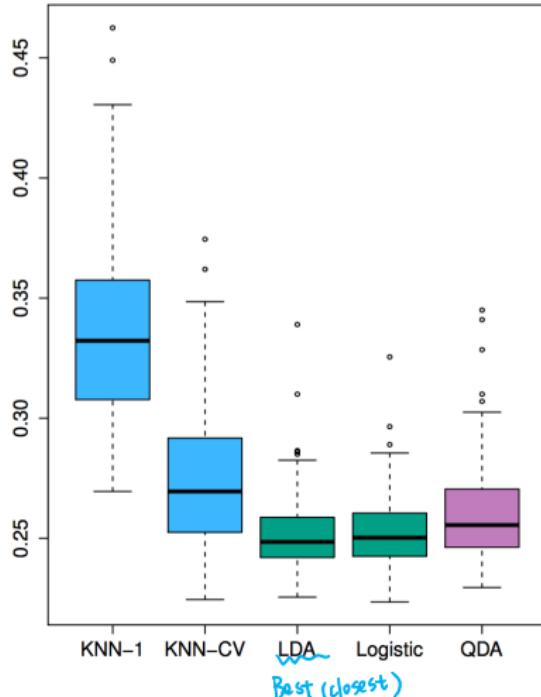
- ▶ e.g. Find the threshold that brings the False negative rate below an acceptable level.

Comparing classification methods through simulation

1. Simulate data from several different known distributions with 2 predictors and a binary response variable.
2. Compare the test error (0-1 loss) for the following methods:
 - ▶ KNN-1
 - ▶ KNN-CV (“optimal” KNN)
 - ▶ Logistic regression
 - ▶ Linear discriminant analysis (LDA)
 - ▶ Quadratic discriminant analysis (QDA)

Scenario 1

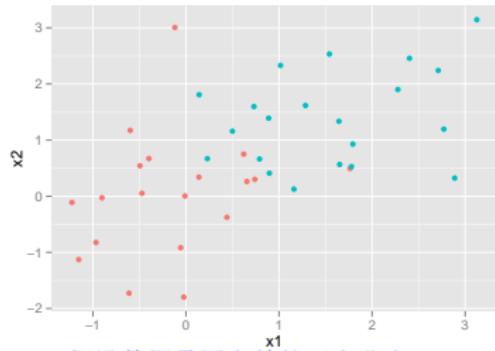
SCENARIO 1



KNN-1 适用于数据关系十分复杂，难以用线性表示的时候

数据看上去线性可分的程度比较高

- ▶ X_1, X_2 standard normal.
- ▶ No correlation in either class.

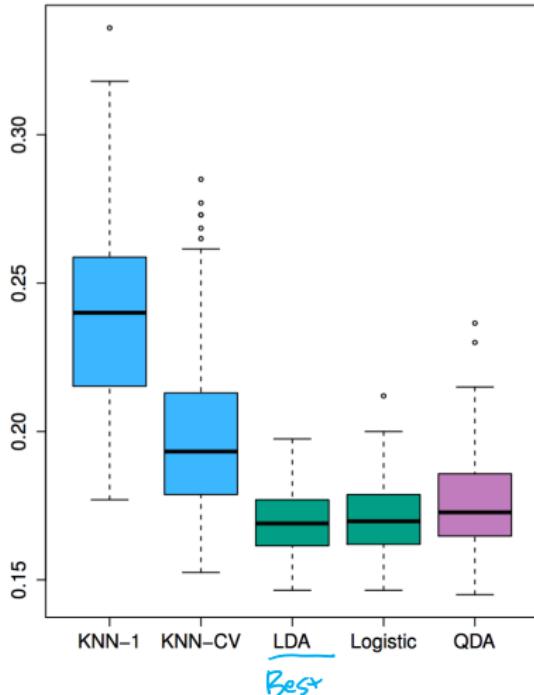


LDA假设数据是同方差的正态分布

logistics模型假设的是数据是线性可分的

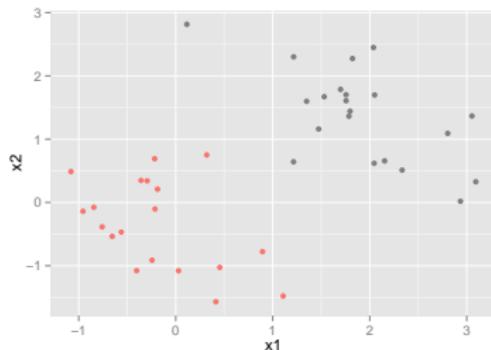
Scenario 2

SCENARIO 2



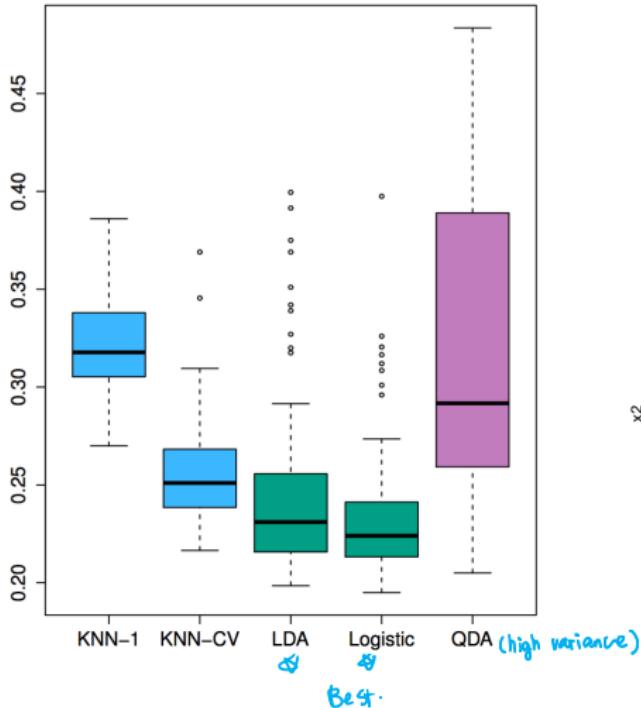
同样线性可分程度比较高

- ▶ X_1, X_2 standard normal.
- ▶ Correlation is -0.5 in both classes.



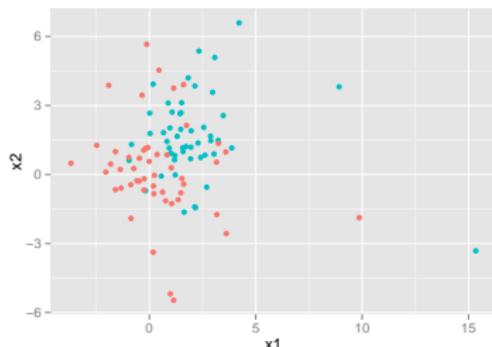
Scenario 3

SCENARIO 3



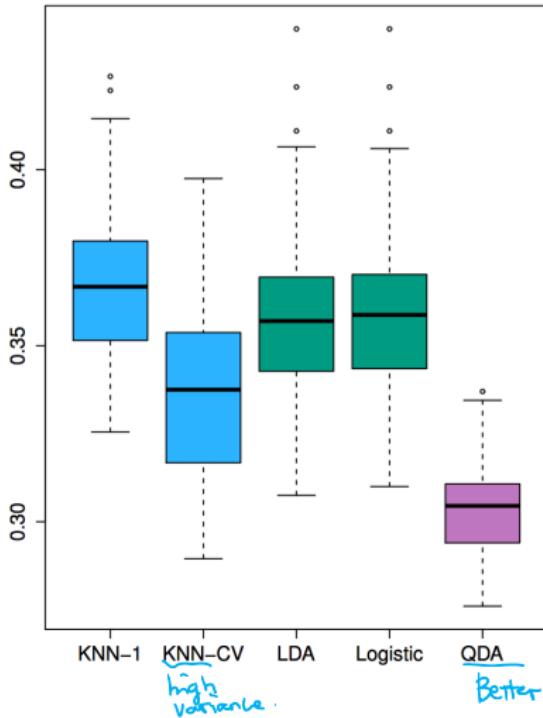
数据本身的方差较大

- X_1, X_2 Student t random variables.
heavier tails
- No correlation in either class.



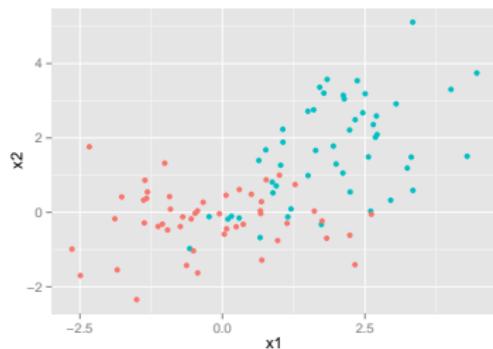
Scenario 4

SCENARIO 4



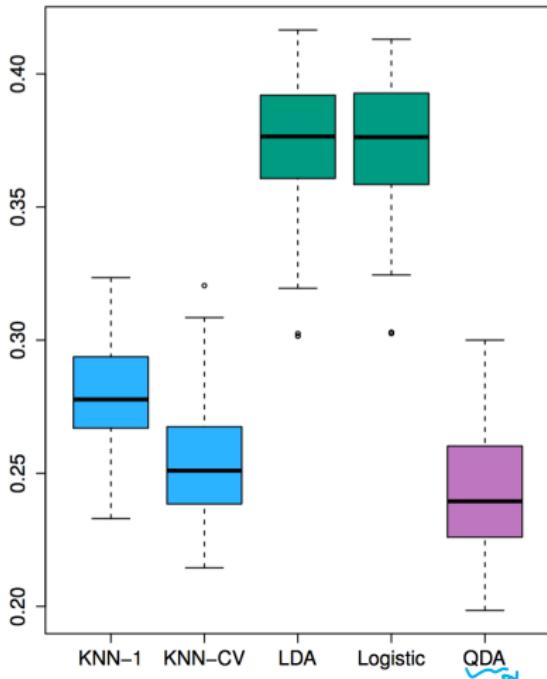
不同的方差

- X_1, X_2 standard normal.
- First class has correlation 0.5, second class has correlation -0.5.



Scenario 5

SCENARIO 5



- ▶ X_1, X_2 uncorrelated, standard normal.
- ▶ Response Y was sampled from:

Satisfy assumption of QDA

$$P(Y = 1|X) = \frac{e^{\beta_0 + \beta_1(X_1^2) + \beta_2(X_2^2) + \beta_3(X_1 X_2)}}{1 + e^{\beta_0 + \beta_1(X_1^2) + \beta_2(X_2^2) + \beta_3(X_1 X_2)}}.$$

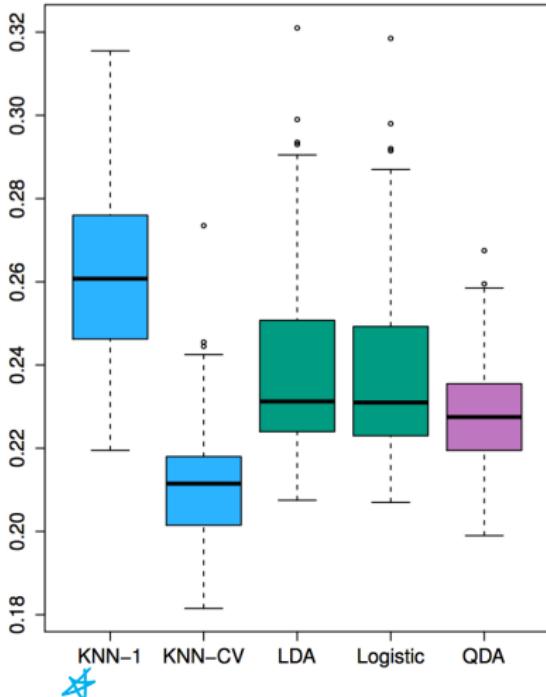
数据的形式是先行不可分的

- ▶ The true decision boundary is quadratic.

二次项形式

Scenario 6

SCENARIO 6



- ▶ X_1, X_2 uncorrelated, standard normal.
- ▶ Response Y was sampled from:

$$P(Y = 1|X) = \frac{e^{f_{\text{nonlinear}}(X_1, X_2)}}{1 + e^{f_{\text{nonlinear}}(X_1, X_2)}}.$$

形式越复杂
KNN的又是越明显
但是KNN在高纬度的情况下表现不佳

- ▶ The true decision boundary is very rough.

Regression I

Professor: Hammou El Barmi
Columbia University

Regression Analysis

- A statistical tool for studying the relationship between one variable (response variable) and other variables (predictor variables)
- Explain the effect of change in a predictor variable on response variable
- Predict the value of response variable based on the value(s) of predictor variable(s)

The response variable is called dependent variable

Predictor variables are called independent variables or explanatory variables

Example

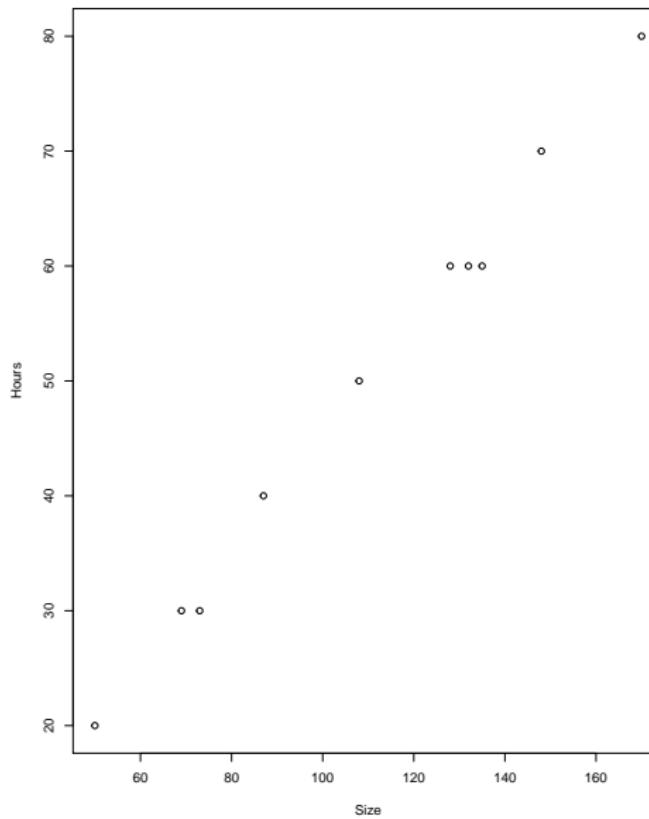
- A company manufactures standard wall clocks
- Wholesalers order the clocks in lot sizes
- The company wants to study relation between lot sizes and man-hours used for manufacture
- Data from a small sample are shown on the next slide

Lot size (X)	Man-hour (Y)
30	73
20	50
60	128
80	170
40	87
50	108
60	135
30	69
70	148
60	132

Example

```
> regdata<-read.table("/Users/HElbarmi/Desktop/EDA/Regressin/Lotsize.txt",head=1)
> regdata
  Size Hours
1    30    73
2    20    50
3    60   128
4    80   170
5    40    87
6    50   108
7    60   135
8    30    69
9    70   148
10   60   132
> Size<-regdata[,2]
> Hours<-regdata[,1]
> plot(Size, Hours, xlab="Hours", ylab="Size")
```

Example



Model is

$$\begin{aligned} Y_{X=x} &= E(Y|X=x) + \epsilon \\ &= \beta_0 + \beta_1 x + \epsilon \end{aligned}$$

That is, we assume that $E(Y|X=x) = \beta_0 + \beta_1 x$

- $\beta_0 = E(Y|X=0)$, β_0 is the mean of Y when $X=0$
- β_1 is the change in the mean of Y corresponding to a one unit in X .

- Suppose our data is $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.

- Therefore

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

- Estimates, b_0 and b_1 of β_0 and β_1 are solution to

$$\min_{\beta_0, \beta_1} \sum_{i=1}^n (y_i - \beta_0 + \beta_1 x_i)^2$$

i.e. they are the values of β_0 and β_1 that solve

$$\min_{\beta_0, \beta_1} \sum_{i=1}^n \epsilon_i^2$$

The solution is

$$\begin{aligned} b_0 &= \bar{y} - b_1 \bar{x} \\ b_1 &= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = r \frac{s_y}{s_x} \end{aligned}$$

Here r is the correlation coefficient

```
> lm(Hours~Size)
Call:
lm(formula = Hours ~Size)
Coefficients:
(Intercept)      Size
          10           2
```

This gives

$$b_0 = 10 \quad \text{and} \quad b_1 = 2$$

The estimated regression line is

$$\widehat{\text{Hours}} = 10 + 2\text{Size}$$

- Here b_0 has no meaningful interpretation
- $b_1 = 2$ means that if increase the size of the lot by one, the number of hours required to do the work will increase by about 2 hours.

```
> summary(lm(Hours~Size))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.00000	2.50294	3.995	0.00398 **
Size	2.00000	0.04697	42.583	1.02e-10 ***

Residual standard error: 2.739 on 8 degrees of freedom

Multiple R-squared: 0.9956, Adjusted R-squared: 0.9951

F-statistic: 1813 on 1 and 8 DF, p-value: 1.02e-10

A $100(1 - \alpha)\%$ confidence interval for β_i is

$$b_i \pm t_{\alpha/2}(n - 2)SE(b_i)$$

```
> confint(lm(Hours~Size))
              2.5 %    97.5 %
(Intercept) 4.228211 15.771789
Size         1.891694  2.108306
```

Interpretation: We are 95% confident that a one unit increase in lot size will increase on average the number of hours required to process the lot by a number between 1.89 hours and 2.11 hours.

- Total Sum of Squares (SST) = Total variation in the response
- Regression Sum of Squares (SSR) = Variation in the response explained by the explanatory (predictor) variable
- Error Sum of Squares (SSE) = Variation in the response not explained by the explanatory (predictor) variable
- $SST = SSR + SSE$ and

$$SST = \sum_{i=1}^n (Y_i - \bar{Y})^2, \quad SSR = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2 \quad \text{and} \quad SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

- The coefficient of determination is

$$R^2 = \frac{SSR}{SST}$$

As a percentage, this is the percentage variability in the response explained by the predictor variable.

- The ANOVA table is given by

Source	df	SS	MS	F
Model	1	SSR	MSR=SSR/1	MSR/MSE
Error	n-2	SSE	MSE=SSE/(n-2)	
Total	n-1	SST		

- The coefficient of determination is

$$R^2 = \frac{SSR}{SST}$$

- MSE is an estimate of σ^2
- To test $H_0 : \beta_1 = 0$ against $H_a : \beta_1 \neq 0$ we reject H_0 if $F > F(1 - \alpha, 1, n - 2)$ or if $p-value < \alpha$.

```
> aov(lm(Hours~Size))
Call:
aov(formula = lm(Hours ~ Size))
```

Terms:

	Hours	Residuals
Sum of Squares	13600	60
Deg. of Freedom	1	8

Residual standard error: 2.738613

Estimated effects may be unbalanced

- $SSR = 13600$, $SSE = 60$ and $SST = SSR + SSE = 13660$
- In addition $R^2 = 13600/13660 = 0.9956$.
- Interpretation: about 99.56% of the variability in the number of hours required to process a lot is explained by its size.

The ANOVA table is

Source	df	SS	MS	F
Model	1	13600	13600	1813.33
Error	8	60	7.5	
Total	9	13660		

```
> summary(lm(Hours~Size))
```

Residual standard error: 2.739 on 8 degrees of freedom

Multiple R-squared: 0.9956, Adjusted R-squared: 0.9951

F-statistic: 1813 on 1 and 8 DF, p-value: 1.02e-10

An estimate of the error variance is $MSE = 7.5$

To test $H_0 : \beta_1 = 0$ against $H_a : \beta_1 \neq 0$ we reject H_0 since $p\text{-value} < 0.05$

- An estimator of μ_x is

$$\hat{y}_x = b_0 + b_1 x$$

- A $100(1 - \alpha)\%$ confidence interval for μ_x is

$$\hat{y}_x \pm t_{n-2}(\alpha/2) \sqrt{MSE} \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

Example

- Example: suppose we want to estimate the average number of hours it will take to process a lot of size equal to 45 using a 95% confidence interval
- In R we use

```
> fit<-lm(Hours~Size)
> predict(fit,newdata = data.frame(Size=45), interval="confidence")
   fit      lwr      upr
100 97.93082 102.0692
```

- The output shows that $\hat{y}_{45} = 100$ and a 95% confidence interval for the average number of hours it will take to process a lot of size 45 is [97.93, 102.07].
- Interpretation: We are 95% confident that on average it will take between 97.93 hours and 102.07 hours to process a lot of size 45.

- A predicted value \hat{y}_x of the response when $X = x$

$$\hat{y}_x = b_0 + b_1 x$$

- A $100(1 - \alpha)\%$ prediction interval for y_x is

$$\hat{y}_x \pm t_{n-2}(\alpha/2) \sqrt{MSE} \sqrt{1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

Example

- Example: suppose we want to estimate the average number of hours it will take to process a lot of size equal to 45 using a 95% confidence interval
- In R we use

```
> fit<-lm(Hours~Size)
> predict(fit,newdata = data.frame(Size=45), interval="prediction")
   fit      lwr      upr
1 100 93.35441 106.6456
```

- The output shows that $\hat{y}_{45} = 100$ and a 95% confidence interval for the average number of hours it will take to process a lot of size 45 is [93.35, 106.65].
- Interpretation: We predict with 95% confident that it will take between 93.35 hours and 106.65 hours to process a lot of size 45.

- Joint estimation of β_0 and β_1
- For the data where $X = 0$ is meaningful, β_0 should be estimated as well as β_1
- If separate 95% confidence intervals for β_0 and β_1 are constructed, respectively, then
 - 5% of the samples would result in a confidence interval of β_0 that does not contain it. Another 5% (possibly the same) will result in a confidence interval of β_1 that does not contain it.
 - Thus, as much as 10% of the samples will result in intervals for β_0 and β_1 that do not contain either or both parameters.
- Joint 95% CIs for $\beta_0 \& \beta_1$ can be constructed to ensure 95% correctness of the entire set of CIs
- Family confidence coefficient: The proportion of samples that are correct for every member of a family of CIs in repeated sampling and recalculation of each CI.

Bonferroni joint CIs

- To achieve $1 - \alpha$ family confidence, each of β_0 and β_1 is estimated with $1 - \alpha/2$ confidence
- $1 - \alpha$ Bonferroni joint CIs are given by

$$b_0 \pm Bs_{b_0} \quad \text{and} \quad b_1 \pm Bs_{b_1}$$

where

$$B = t_{n-2}(\alpha/4)$$

- $1 - \alpha$ is a lower bound on the true family confidence coefficient
- Family confidence coefficients are often specified at lower levels (say 90%)

```
> fit<-lm(Hours~Size)
> confint(fit, level=1-0.05/4)
              0.625 % 99.375 %
(Intercept) 1.975688 18.024312
Size          1.849426  2.150574
```

Working Hotelling Method

- works for unlimited family of CIs
- as a result, yields a confidence band for the entire response line
- it is given by

$$\hat{y}_x \pm W \sqrt{MSE} \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

where

$$W^2 = 2F(1 - \alpha, 2, n - 2)$$

- When the number of CIs is small, Bonferroni method gives smaller widths than Working-Hotelling method generally

Simultaneous estimation of mean response

```
ci.wh <- function(fit, newdata, alpha = 0.1){  
  df      <- nrow(model.frame(fit)) - length(coef(fit))  
  W       <- sqrt( 2 * qf(1 - alpha, length(coef(fit)), df) )  
  ci      <- predict(fit, newdata, se.fit = TRUE)  
  x <- cbind(  
    'x'   = newdata,  
    's'   = ci$se.fit,  
    'fit' = ci$fit,  
    'lwr' = ci$fit - W * ci$se.fit,  
    'upr' = ci$fit + W * ci$se.fit)  
  
  return(x)  
}  
  
newdata<-data.frame(Size=c(45,65,75))  
ci.wh(fit,newdata)
```

Simultaneous estimation of mean response

	Size	s	fit	lwr	upr
1	45	0.8972999	100	97.82992	102.1701
2	65	1.1163886	140	137.30006	142.6999
3	75	1.4589984	160	156.47148	163.5285

Recall that the model is

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

and we assumed that

- ϵ_i is normally distributed with mean zero and variance σ^2
- ϵ_i s are independent

If these assumptions do not hold, they invalidate our analysis

Diagnostics:

- Examine appropriateness of model & and detect violations of model assumptions
- Typical violations
 - Regression function is not linear
 - Error terms do not have constant variance
 - Error terms are not independent
 - One or more observations are outliers
 - Error terms are not normally distributed
 - One or more important predictors have been omitted from the model

We use the residuals to examine important departures from the simple linear regression model

$$y = \beta_0 + \beta_1 x + \epsilon \quad \text{with independent and identically normally distributed errors}$$

- The i th residual $e_i = y_i - \hat{y}_i, i = 1, 2, \dots, n.$
- The residuals are used to estimate the errors
- $\sum_{i=1}^n e_i = 0.$
- $\text{var}(e_i) = \sigma^2(1 - h_{ii})$

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

- The residuals are not independent

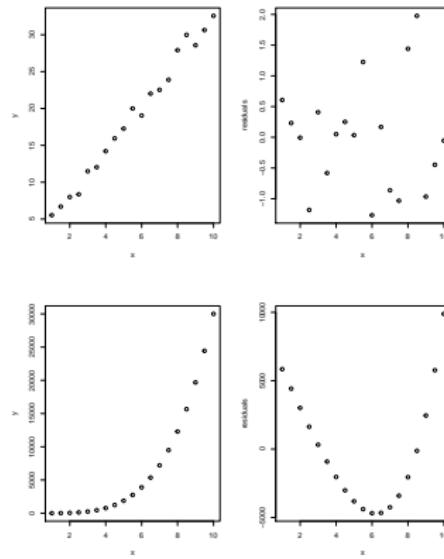
We use plots of the residuals to answer these questions. The plots that are commonly used are

- ① plot the residuals against the predictor variable
- ② plot the residuals against the fitted values
- ③ plot the residuals against the time (important if data collected over time)
- ④ plot the residuals against omitted variables
- ⑤ box plot for the residuals
- ⑥ normal plot for the residuals

We should check for:

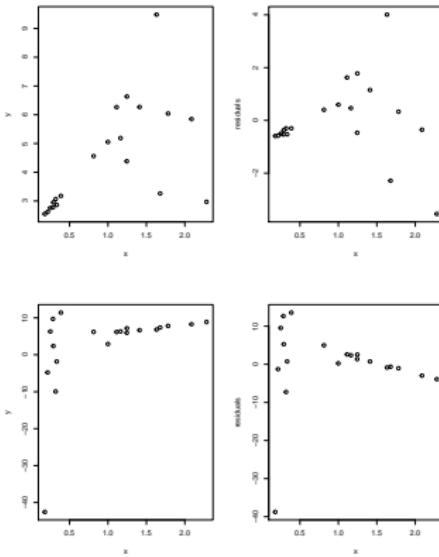
1. Nonlinearity of the regression function: this can be studied by a plot of the residuals against the predictor variable or equivalently by a plot of the residuals against the fitted values. The plot should not show any particular pattern.

Figure: Linear in top not linear in the bottom



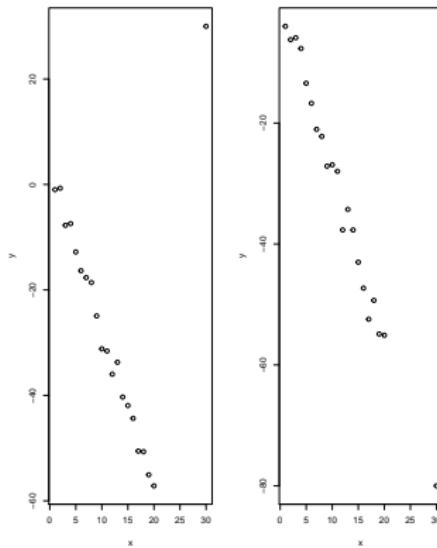
2. Nonconstancy of the error variance: plot residuals versus the predictor variable or the fitted values

Figure: Noconstant Variance



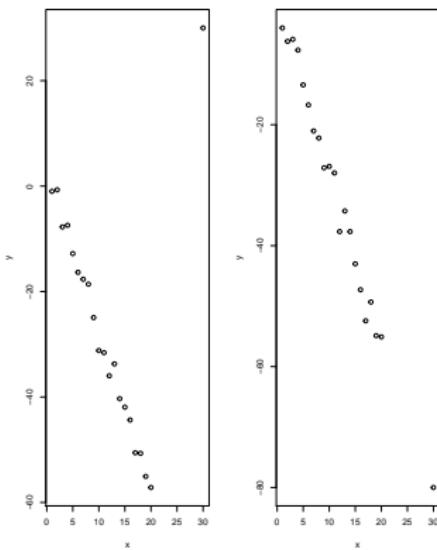
3. Presence of outliers: outliers are extreme observations. An outlier may dramatically change the regression line (when this is the case the outliers in an influential case). Outlier residuals can be identified from residual plots of residuals versus x or \hat{y} as well as box plots.

Figure: Presence of outliers



4. Nonindependent errors: plot residuals versus time to see if there is any cyclical pattern. The residuals are always dependent but this dependency decreases with the sample size.

Figure: Presence of outliers



5. Nonnormality of error terms: make a box plot of the residuals

Regression II

Professor: Hammou El Barmi
Columbia University

$$Y_i = \beta_0 + \beta_1 X_{1i} + \dots + \beta_p X_{pi} + \epsilon_i$$

Here

- β_0 is the mean of Y when all the Xs are equal to zero
- β_i is the change in the mean of Y when we increase X_i by one while holding all the other Xs fixed

In matrix formulation,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{21} & \dots & x_{p1} \\ 1 & x_{12} & x_{22} & \dots & x_{p2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & \dots & x_{pn} \end{pmatrix}$$

Example: antique grandfather clocks

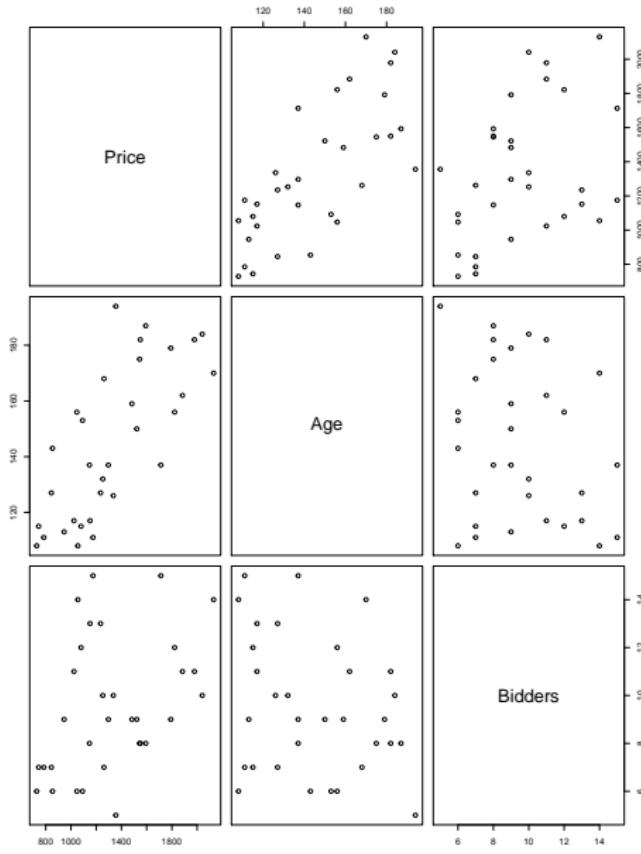
- The data give the selling price at auction of 32 antique grandfather clocks. Also recorded is the age of the clock and the number of people who made a bid.
- The variables are
 - Age : Age of the clock (years)
 - Bidders: Number of individuals participating in the bidding
 - Price: Selling price (pounds sterling)

	Age	Bidders	Price
1	127	13	1235
2	115	12	1080
3	127	7	845
4	150	9	1522
5	156	6	1047
6	182	11	1979
7	156	12	1822
8	132	10	1253
9	137	9	1297

Example: antique grandfather clocks

10	113	9	946
11	137	15	1713
12	117	11	1024
13	137	8	1147
14	153	6	1092
15	117	13	1152
16	126	10	1336
17	170	14	2131
18	182	8	1550
19	162	11	1884
20	184	10	2041
21	143	6	854
22	159	9	1483
23	108	14	1055
24	175	8	1545
25	108	6	729
26	179	9	1792
27	111	15	1175
28	187	8	1593
29	111	7	785
30	115	7	744
31	194	5	1356
32	168	7	1262

Example (Clocks continued)



To estimate β we minimize

$$\sum_{i=1}^n \epsilon_i^2 = \epsilon^T \epsilon = (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta)$$

The solution is

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

Under the assumption we made before

$$\mathbf{b} \sim N(\beta, \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1})$$

This implies in particular that

- $E(\mathbf{b}) = \beta$, that is, \mathbf{b} is an unbiased estimator of β .
- $Var(\mathbf{b}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$ and is estimated by

$$Var(\mathbf{b}) = MSE(\mathbf{X}^T \mathbf{X})^{-1}$$

- An estimate of the variance of β_i is

$$SE(b_i) = MSE(\mathbf{X}^T \mathbf{X})_{ii}^{-1}$$

where $(\mathbf{X}^T \mathbf{X})_{ii}^{-1}$ is the i th diagonal element of $(\mathbf{X}^T \mathbf{X})^{-1}$.

```
> attach(data)
> fit<-lm(Price~ Age+Bidders)
> fit
```

Call:

```
lm(formula = Price ~ Age + Bidders)
```

Coefficients:

(Intercept)	Age	Bidders
-1336.72	12.74	85.82

The regression equation is

$$\widehat{\text{Price}} = -1336.72 + 12.74\text{Age} + 85.82\text{Bidders}$$

- A $100(1 - \alpha)\%$ confidence interval for β_i is

$$b_i \pm t_{n-p-1}(\alpha/2)SE(b_i)$$

The interpretation of this confidence interval is: We are $100(1 - \alpha)\%$ confident that when we increase X_i by one unit while holding all the other Xs fixed, the average, Y changes by an amount in this interval.

```
> confint(fit)
              2.5 %      97.5 %
(Intercept) -1691.27514 -982.16896
Age          10.89062   14.58177
Bidders     68.00986  103.62040
```

We are 95% that when we increase age by one year while holding the number of bidders fixed, on average the price goes by an amount between 10.89 and 12.58 pounds sterling.

Analysis of Variance Approach

To test $H_0 : \beta_i = \beta_{i0}$ against $H_a : \beta_i \neq \beta_{i0}$, the test statistic is

$$t = \frac{b_i - \beta_{i0}}{SE(b_i)}$$

and we reject H_0 if

$$|t| > t_{n-p-1}(\alpha/2) \text{ or if } p\text{-value} < \alpha$$

```
> summary(fit)
lm(formula = Price ~ Age + Bidders)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1336.7221	173.3561	-7.711	1.67e-08 ***
Age	12.7362	0.9024	14.114	1.60e-14 ***
Bidders	85.8151	8.7058	9.857	9.14e-11 ***

Residual standard error: 133.1 on 29 degrees of freedom

Multiple R-squared: 0.8927, Adjusted R-squared: 0.8853

F-statistic: 120.7 on 2 and 29 DF, p-value: 8.769e-15

p-values very small we reject $H_0 : \beta_i = 0$ against $H_a : \beta_i \neq 0$

- The ANOVA table is given by

Source	df	SS	MS	F
Model	p	SSR	MSR=SSR/p	MSR/MSE
Error	n-p-1	SSE	MSE=SSE/(n-p-1)	
Total	n-1	SST		

- The coefficient of determination is

$$R^2 = \frac{SSR}{SST}$$

- Adjusted

$$R_{adj}^2 = 1 - \frac{n-1}{n-p-1} \frac{SSE}{SST}$$

can be used for model selection

- MSE is an estimate of σ^2

- To test $H_0 : \beta_1 = \beta_2 = \dots = \beta_p = 0$ against $H_a : \text{at least one of these } \beta\text{s is not zero}$, we reject H_0 if $F > F(1 - \alpha, p, n - p - 1)$ or if $p - \text{value} < \alpha$.
- The test statistic is given by

$$F = \frac{(SSE_R - SSE_F)/(df_R - df_F)}{SSE_F/df_F}$$

and we reject H_0 if

$$F > F(1 - \alpha, df_R - df_F, df_F)$$

or if $p - \text{value} < \alpha$.

- In the example, to test $H_0 : \beta_1 = \beta_2 = 0$ against $H_a : \text{at least one of them is not equal to zero}$, we

F-statistic: 120.7 on 2 and 29 DF, p-value: 8.769e-15

Since the p-value is very small we reject H_0

- Suppose we want to test $H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0, k < p$ against $H_a : \text{Not } H_0$.
- In this case we have two models:
 - a reduced model(the model in which $\beta_1 = \beta_2 = \dots = \beta_k = 0$) and
 - a full model in which we have all the β s

- we have up to this point assumed that $y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, 2, \dots, n$, ϵ_i s are iid $N(0, \sigma^2)$ and made inference about β_0 and β_1
- The goal of the lack of fit test is to test a specific type of regression function fits the data.
- The lack fit test assumes that the observations for a given x are
 - ① independent of each other
 - ② normally distributed
 - ③ the distribution of y given x have the same variance σ^2 .
- We want to test

$$H_0 : \mu_x = \beta_0 + \beta_1 x$$

$$H_1 : \mu_x \neq \beta_0 + \beta_1 x$$

- To carry out a lack of fit test requires repeat observations at one or more x levels

Lack of fit test

data

x	y	mean under H_0	mean under H_a
x_1	$y_{11}, y_{12}, \dots, y_{1n_1}$	$\beta_0 + \beta_1 x_1$	μ_{x_1}
x_2	$y_{21}, y_{22}, \dots, y_{2n_2}$	$\beta_0 + \beta_1 x_2$	μ_{x_2}
\vdots	\vdots	\vdots	\vdots
x_c	$y_{c1}, y_{c2}, \dots, y_{cn_c}$	$\beta_0 + \beta_1 x_c$	μ_{x_2}

- Under H_a , the model is $y_{ij} = \mu_i + \epsilon_{ij}$, $i = 1, 2, \dots, c, j = 1, 2, \dots, n_i$
- The estimate of μ_i is $\bar{y}_i = \frac{\sum_{j=1}^{n_i} y_{ij}}{n_i}$
- $SSE_R = \sum_{i=1}^c \sum_{j=1}^{n_i} (y_{ij} - \hat{y}_{ij})^2$ where $\hat{y}_{ij} = b_0 + b_1 x_i$. H0对应的是reduced model
- $SSE_F = \sum_{i=1}^c \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2$ H1 对应的是full model
- The partial F-test is

$$F = \frac{SSE_R - SSE_F}{df_R - df_F} \div \frac{SSE_F}{df_F}$$

- Reject H_0 if $F > F(1 - \alpha, df_R - df_F, df_F)$ or if $p-value < \alpha$
- $df_F = \sum_{i=1}^c n_i - c$ and $df_R = \sum_{i=1}^c n_i - 2$
- The difference $SSE_R - SSE_F$ is called the lack of fit sum of squares and is denoted by SSLF
- The test statistic is sometimes expressed as

$$F = \frac{MSLF}{MSE}$$

Lack of fit test

x	y
0.01	127.6
0.48	124.0
0.71	110.8
0.95	103.9
1.19	101.5
0.01	130.1
0.48	122.0
1.44	92.3
0.71	113.1
1.96	83.7
0.01	128.0
1.44	91.4
1.96	86.2

```
> Reduced <- lm(y ~ x, data=corrosion)
> summary(Reduced)

Coefficients:
            Estimate Std. Error t value  Pr(>|t|)    
(Intercept) 129.79    1.40     92.5   < 2e-16  
x           -24.02    1.28    -18.8   1.1e-09  
Residual standard error: 3.06 on 11 degrees of freedom
Multiple R-Squared: 0.97, Adjusted R-squared: 0.967 
F-statistic: 352 on 1 and 11 degrees of freedom, p-value: 1.06e-09
```

Lack of fit test

```
>Full <- lm(y~factor(x))
> summary(lm(Full))
Call:
lm(formula = y ~ factor(x))
```

Residuals:

Min	1Q	Median	3Q	Max
-1.2500	-0.9667	0.0000	1.0000	1.5333

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	128.567	0.809	158.914	4.19e-12 ***
factor(x)0.48	-5.567	1.279	-4.352	0.00481 **
factor(x)0.71	-16.617	1.279	-12.990	1.28e-05 ***
factor(x)0.95	-24.667	1.618	-15.245	5.03e-06 ***
factor(x)1.19	-27.067	1.618	-16.728	2.91e-06 ***
factor(x)1.44	-36.717	1.279	-28.703	1.18e-07 ***
factor(x)1.96	-43.617	1.279	-34.097	4.24e-08 ***

Signif. codes:	0 *** 0.001 ** 0.01 * 0.05 . 0.1 1			

Residual standard error: 1.401 on 6 degrees of freedom

Multiple R-squared: 0.9965, Adjusted R-squared: 0.9931

F-statistic: 287.3 on 6 and 6 DF, p-value: 4.152e-07

To test for Lack of fit, we use

```
> anova(Reduced,Full)
Analysis of Variance Table
Model 1: y ~ x
Model 2: y ~ factor(x)
Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     11 102.9
2      6 11.8      5    91.1  9.28   0.0086
```

$$SSE_R = 102.9, SSE_F = 11.8, df_R = 11, df_F = 6.$$

$$F = \frac{102.9 - 11.8}{11 - 6} \div \frac{11.8}{6} = \frac{91.1}{5} \div \frac{11.8}{6} = 9.28.$$

The p-value is 0.0086. Reject $H_0 : \mu_x = \beta_0 + \beta_1 x$.

- Y = volume of sales in July of some electronic store
- x = number of households in the location
- Location of the store = $\begin{cases} \text{Mall} \\ \text{Downtown} \\ \text{Street} \end{cases}$

number of household	location	sales
161	street	157.27
99	street	93.28
135	street	136.81
120	street	123.79
164	street	153.51
221	mall	241.74
179	mall	201.54
204	mall	206.71
214	mall	229.78
101	mall	135.22
231	downtown	224.71
206	downtown	195.29
248	downtown	242.16
107	downtown	115.21
205	downtown	197.82

```
> fit
Call:
lm(formula = sales ~ nhousehold + factor(location))

Coefficients:
              (Intercept)          nhousehold    factor(location)mall
                           21.8415                  0.8686                  21.5100
factor(location)street
                           -6.8638
```

Regression with qualitative variables

```
> summary(fit)
```

Call:

```
lm(formula = sales ~ nhousehold + factor(location))
```

Residuals:

Min	1Q	Median	3Q	Max
-13.834	-2.999	2.225	4.357	6.431

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	21.84147	8.55848	2.552	0.026898 *
nhousehold	0.86859	0.04049	21.452	2.52e-10 ***
factor(location)mall	21.50998	4.06509	5.291	0.000256 ***
factor(location)street	-6.86378	4.77048	-1.439	0.178047

Residual standard error: 6.349 on 11 degrees of freedom

Multiple R-squared: 0.9868, Adjusted R-squared: 0.9833

F-statistic: 275.1 on 3 and 11 DF, p-value: 1.268e-10

```
> confint(fit)
                2.5 %      97.5 %
(Intercept)    3.0043933 40.6785468
nhousehold     0.7794707  0.9577061
factor(location)mall 12.5627722 30.4571864
factor(location)street -17.3635248  3.6359712
```

- Multicollinearity: it exists when the explanatory variables are linearly dependent.
- We use the variance inflation factor (VIF) to check whether or not multicollinearity exists
- The VIF for variable x_j is

$$VIF = \frac{1}{1 - R_j^2}$$

where R_j^2 is the coefficient of determination when x_j is regressed on the other x_i s

- As a percentage, R_j^2 is the percentage variability in x_j explained by the other x_i s
- It turns out that $SE(b_j) \propto \sqrt{VIF}$, so when R_j^2 is high, the $SE(b_j)$ is also large and that leads to failing to reject $H_0 : \beta_j = 0$

- An outlier is a data point whose response y does not follow the general trend of the rest of the data.
- A data point has high leverage if it has "extreme" predictor x values. With a single predictor, an extreme x value is simply one that is particularly high or low. With multiple predictors, extreme x values may be particularly high or low for one or more predictors, or may be "unusual" combinations of predictor values (e.g., with two predictors that are positively correlated, an unusual combination of predictor values might be a high value of one predictor paired with a low value of the other predictor).
- A data point is influential if it unduly influences any part of a regression analysis, such as the predicted responses, the estimated slope coefficients, or the hypothesis test results.
- Outliers and high leverage data points have the potential to be influential, but we generally have to investigate further to determine whether or not they are actually influential
- One advantage of the case in which we have only one predictor is that we can look at simple scatter plots in order to identify any outliers and influential data points.

Outliers and Influential Points

- The hat matrix is

$$H = X(X^T X)^{-1} X^T$$

Note that $\hat{y} = H^T y$ so that

$$\begin{aligned}\hat{y}_1 &= h_{11}y_1 + h_{12}y_2 + \dots + h_{1n}y_n \\ \hat{y}_2 &= h_{21}y_1 + h_{22}y_2 + \dots + h_{2n}y_n \\ &\vdots \quad \vdots \quad \vdots \\ \hat{y}_n &= h_{n1}y_1 + h_{n2}y_2 + \dots + h_{nn}y_n\end{aligned}$$

- h_{ii} is the i th element of the diagonal of H . It measures the distance of the x values of the i th case from the center of the experimental region (ignores the response)
- The leverage h_{ii} , quantifies the influence that the observed response y_i has on its predicted value \hat{y}_i . That is, if h_{ii} is small, then the observed response y_i plays only a small role in the value of the predicted response \hat{y}_i .
- On the other hand, if h_{ii} is large, then the observed response y_i plays a large role in the value of the predicted response \hat{y}_i . It's for this reason that the h_{ii} are called the leverages.
- In simple linear regression

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Some important properties of the leverages:

- ① The leverage h_{ii} is a measure of the distance between the x value for the i th data point and the mean of the x values for all n data points.
- ② The leverage h_{ii} is a number between 0 and 1, inclusive.
- ③ The sum of the h_{ii} equals p , the number of parameters (regression coefficients including the intercept).

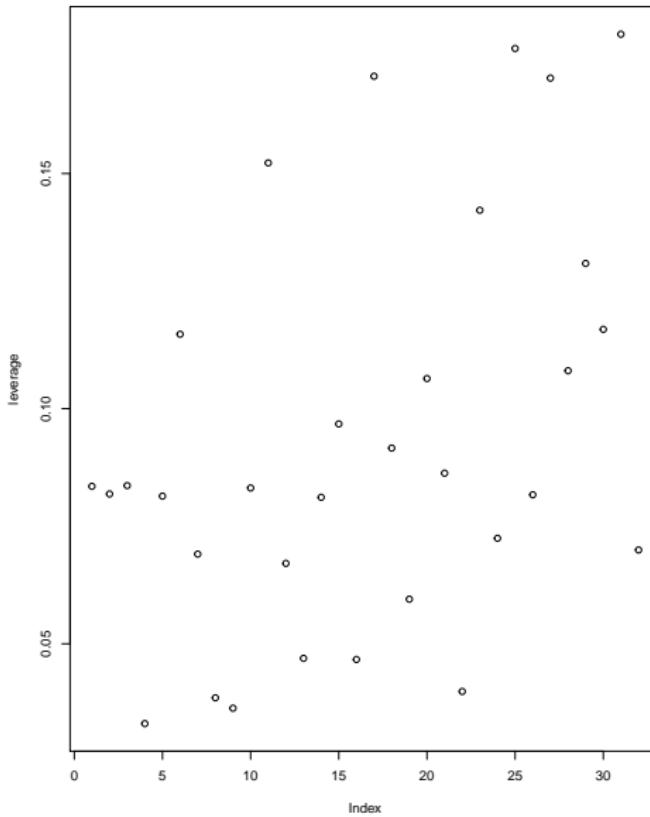
The first bullet indicates that the leverage h_{ii} quantifies how far away the i th x value is from the rest of the x values. If the i th x value is far away, the leverage h_{ii} will be large; and otherwise not.

If the x is far away then the corresponding y will play an important influence in predicted value of y

- The great thing about leverages is that they can help us identify x values that are extreme and therefore potentially influential on our regression analysis.
- How? All we need to do is determine when a leverage value should be considered large. A common rule is to flag any observation whose leverage value, h_{ii} , satisfies

$$h_{ii} > \frac{2p}{n}$$

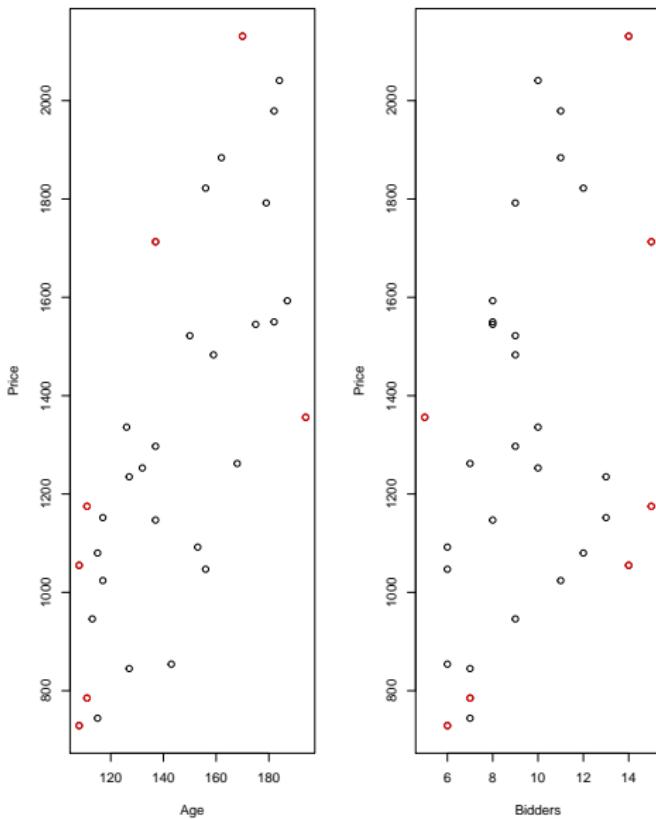
Example (Clocks continued)



Example (Clocks continued)

```
> data[leverage>2*2/32,]  
  Age Bidders Price  
11 137        15  1713  
17 170        14  2131  
23 108        14  1055  
25 108         6   729  
27 111        15  1175  
29 111         7   785  
31 194        5   1356
```

Example (Clocks continued)



- Residuals: The i th residual is defined as $e_i = y_i - \hat{y}_i, i = 1, 2, \dots, n.$
- Studentized residuals (or internally studentized residuals) are defined for each observation, $i = 1, \dots, n$ as an ordinary residual divided by an estimate of its standard deviation:

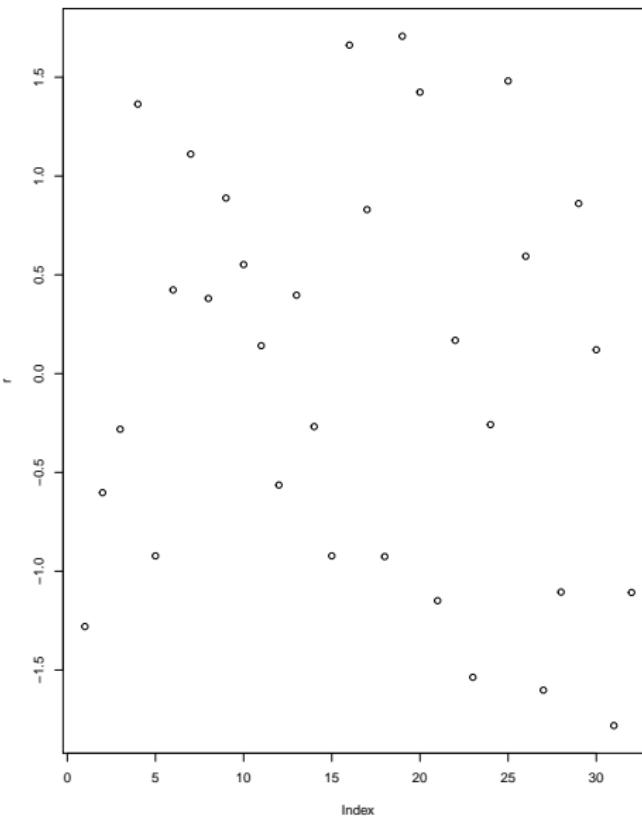
$$r_i = \frac{e_i}{\sqrt{MSE(1 - h_{ii})}}$$

ei 的均值为0, 方差为mse
(1-leverage)

- An observation with an internally studentized residual that is larger than 3 (in absolute value) is generally deemed an outlier. [Sometimes, the term "outlier" is reserved for an observation with an externally studentized residual that is larger than 3 in absolute value? we consider externally studentized residuals in the next section.]

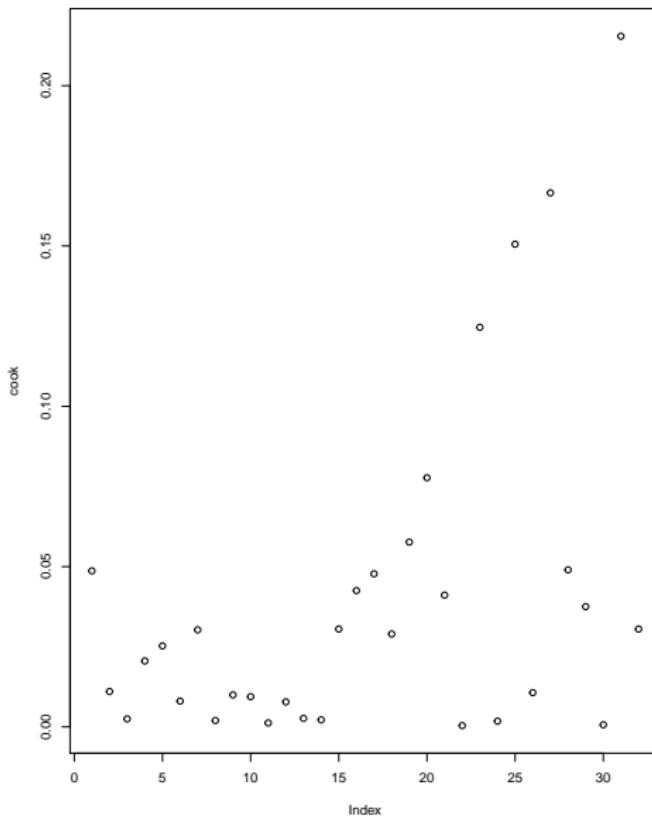
```
> r = rstudent(fit)
> data[abs(r)>3,]
[1] Age      Bidders Price
<0 rows> (or 0-length row.names)
> plot(r)
```

Example (Clocks continued)



- An influential point is one if removed from the data would significantly change the fit.
- An influential point may either be an outlier or have large leverage, or both, but it will tend to have at least one of those properties.
- Cook's distance is a commonly used influence measure that combines these two properties. It can be expressed as
- Typically, points with cook's distance greater than 1 are classified as being influential.
- We can compute the Cook's distance using the following commands
`cook = cooks.distance(fit)`

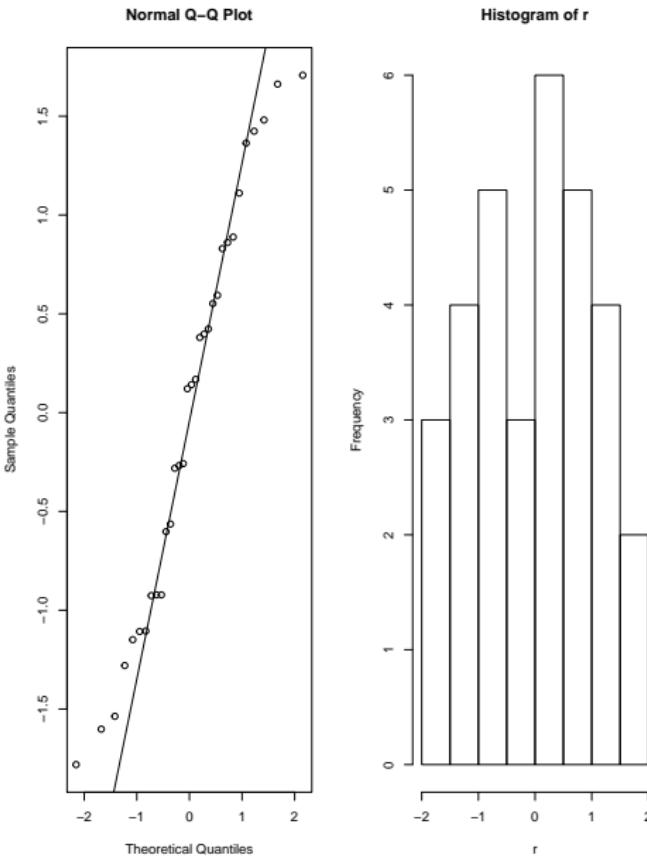
Example (Clocks continued)



- A Normal probability plot of the residuals can be used to check the normality assumption.
- Here each residual is plotted against its expected value under normality. To make normal probability plots , as well as a histogram, type:

```
> qqnorm(fit$res)
> qqline(fit$res)
> hist(fit$res)
```

Example (Clocks continued)



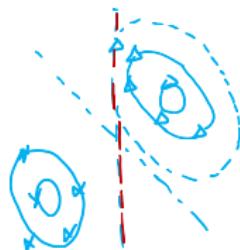
Lecture 7: Support Vector Machines I

Reading: Section 12.2

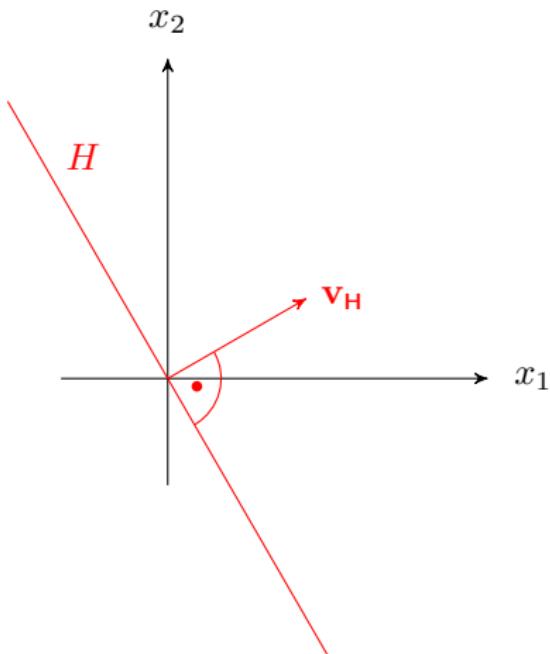
GU4241/GR5241 Statistical Machine Learning

Linxi Liu

February 9, 2018



Hyperplanes



Hyperplanes

A **hyperplane** in \mathbb{R}^d is a linear subspace of dimension $(d - 1)$.

- ▶ A \mathbb{R}^2 -hyperplane is a line, a \mathbb{R}^3 -hyperplane is a plane.
- ▶ As a linear subspace, a hyperplane always contains the origin.

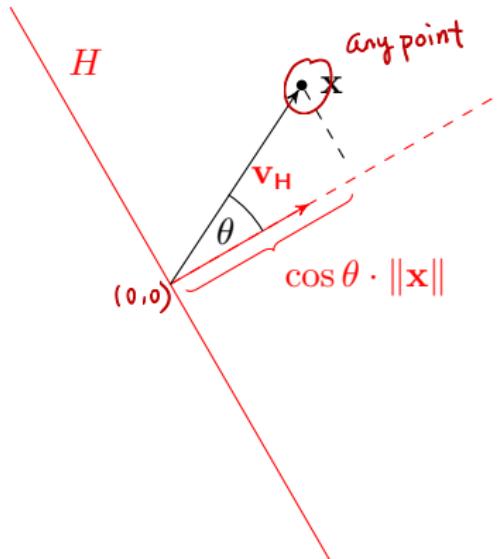
Normal vectors

A hyperplane H can be represented by a **normal vector**. The hyperplane with normal vector \mathbf{v}_H is the set

$$H = \{\mathbf{x} \in \mathbb{R}^d \mid \langle \mathbf{x}, \mathbf{v}_H \rangle = 0\} .$$

*collection of vectors orthogonal to \mathbf{v}_H
inner product is 0*

Which side of the plane are we on?



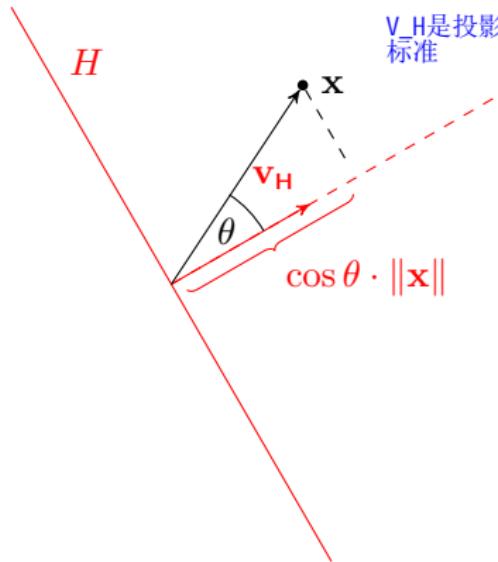
- ▶ The projection of x onto the direction of v_H has length $\langle x, v_H \rangle$ measured in units of v_H , i.e. length $\langle x, v_H \rangle / \|v_H\|$ in the units of the coordinates.
- ▶ Recall the cosine rule for the scalar product,

$$\cos \theta = \frac{\langle x, v_H \rangle}{\|x\| \cdot \|v_H\|} .$$

$(\cos \theta > 0) \quad \theta \in (-\frac{\pi}{2}, \frac{\pi}{2}) \Rightarrow$ ^{x on the} right side of hyperplane
 $(\cos \theta < 0) \quad \theta \in (\frac{\pi}{2}, \frac{3\pi}{2}) \Rightarrow$ ^{x on the} left side ..

If $\langle x, v_H \rangle > 0$ then $y=+1$ (right side)
 < 0 then $y=-1$ (left side)

Which side of the plane are we on?



v_H 是投影长度的
标准

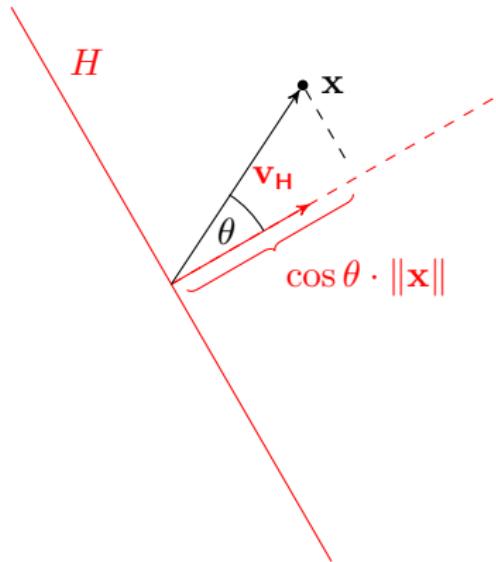
- The projection of \mathbf{x} onto the direction of \mathbf{v}_H has length $\langle \mathbf{x}, \mathbf{v}_H \rangle$ measured *in units of \mathbf{v}_H* , i.e. length $\langle \mathbf{x}, \mathbf{v}_H \rangle / \|\mathbf{v}_H\|$ in the units of the coordinates.
- Recall the cosine rule for the scalar product,

$$\cos \theta = \frac{\langle \mathbf{x}, \mathbf{v}_H \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{v}_H\|} .$$

- Consequence: The **distance** of \mathbf{x} from the plane is given by

$$d(\mathbf{x}, H) = \frac{\langle \mathbf{x}, \mathbf{v}_H \rangle}{\|\mathbf{v}_H\|} = \cos \theta \cdot \|\mathbf{x}\| .$$

Which side of the plane are we on?



- ▶ The projection of \mathbf{x} onto the direction of \mathbf{v}_H has length $\langle \mathbf{x}, \mathbf{v}_H \rangle$ measured in units of \mathbf{v}_H , i.e. length $\langle \mathbf{x}, \mathbf{v}_H \rangle / \|\mathbf{v}_H\|$ in the units of the coordinates.
- ▶ Recall the cosine rule for the scalar product,

$$\cos \theta = \frac{\langle \mathbf{x}, \mathbf{v}_H \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{v}_H\|} .$$

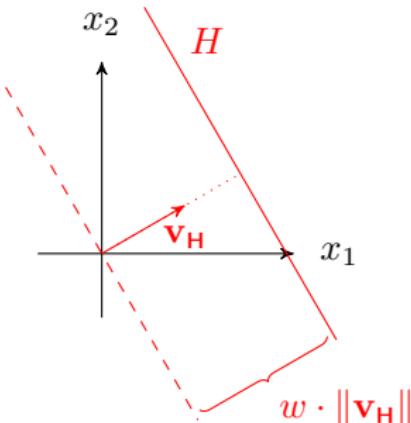
- ▶ Consequence: The distance of \mathbf{x} from the plane is given by

$$d(\mathbf{x}, H) = \frac{\langle \mathbf{x}, \mathbf{v}_H \rangle}{\|\mathbf{v}_H\|} = \cos \theta \cdot \|\mathbf{x}\| .$$

- ▶ We can decide which side of the plane \mathbf{x} is on using

$$\operatorname{sgn}(\cos \theta) = \operatorname{sgn} \langle \mathbf{x}, \mathbf{v}_H \rangle .$$

Affine Hyperplanes

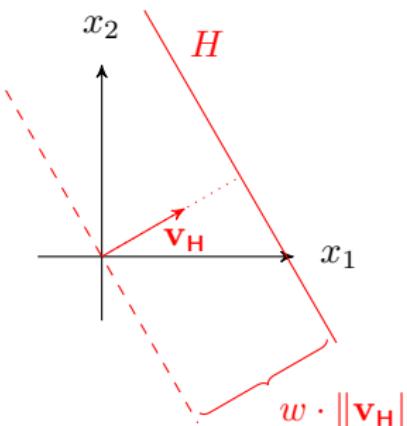


Affine Hyperplanes

- ▶ An **affine hyperplane** H_w is a **hyperplane** translated (**shifted**) by a vector \mathbf{w} , i.e.
$$H_w = H + \mathbf{w}.$$
- ▶ We choose \mathbf{w} in the direction of \mathbf{v}_H , i.e.
$$\mathbf{w} = c \cdot \mathbf{v}_H \text{ for } c > 0.$$

位移的长度是由 c 决定的，方向是由 \mathbf{v}_H 决定的

Affine Hyperplanes



Affine Hyperplanes

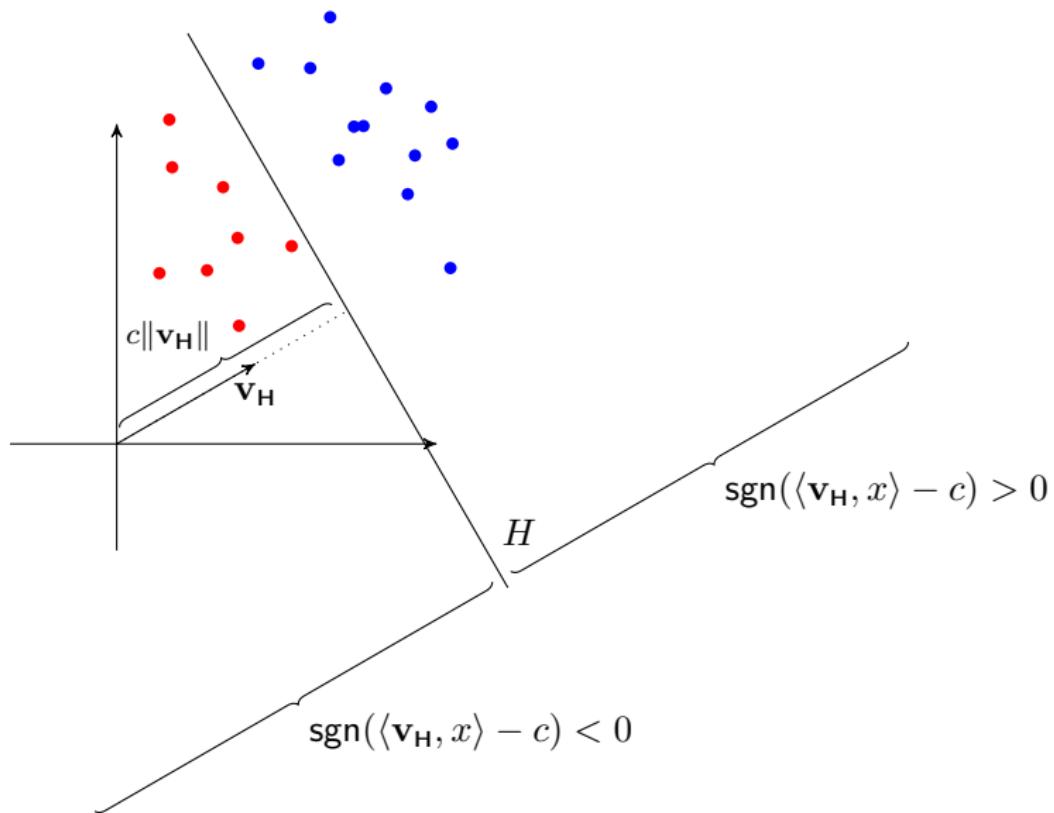
- ▶ An **affine hyperplane** H_w is a hyperplane translated (shifted) by a vector w , i.e.
$$H_w = H + w.$$
- ▶ We choose w in the direction of v_H , i.e.
$$w = c \cdot v_H \text{ for } c > 0.$$

Which side of the plane?

- ▶ Which side of H_w a point x is on is determined by
- ▶ $\text{sgn}(\langle x - w, v_H \rangle) = \text{sgn}(\langle x, v_H \rangle - c \langle v_H, v_H \rangle) = \text{sgn}(\langle x, v_H \rangle - c \|v_H\|^2)$.
look at the sign of $\langle x-w, v_H \rangle$. Larger than 0 \Rightarrow RHS
- ▶ If v_H is a unit vector, we can use

$$\text{sgn}(\langle x - w, v_H \rangle) = \text{sgn}(\langle x, v_H \rangle - c).$$

Classification with Affine Hyperplanes



Linear Classifiers

Definition

A **linear classifier** is a function of the form

这里的內容和感知器是一样的

$$f_H(\mathbf{x}) := \text{sgn}(\langle \mathbf{x}, \mathbf{v}_H \rangle - c),$$

where $\mathbf{v}_H \in \mathbb{R}^d$ is a vector and $c \in \mathbb{R}_+$.

Note: We usually assume \mathbf{v}_H to be a unit vector. If it is not, f_H still defines a linear classifier, but c describes a shift of a different length.

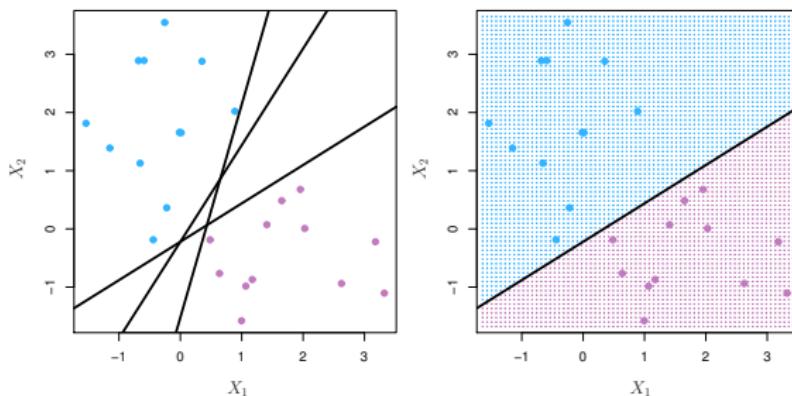
Definition

Two sets $A, B \in \mathbb{R}^d$ are called **linearly separable** if there is an affine hyperplane H which separates them, i.e. which satisfies

$$\langle \mathbf{x}, \mathbf{v}_H \rangle - c = \begin{cases} < 0 & \text{if } \mathbf{x} \in A \\ > 0 & \text{if } \mathbf{x} \in B \end{cases}$$

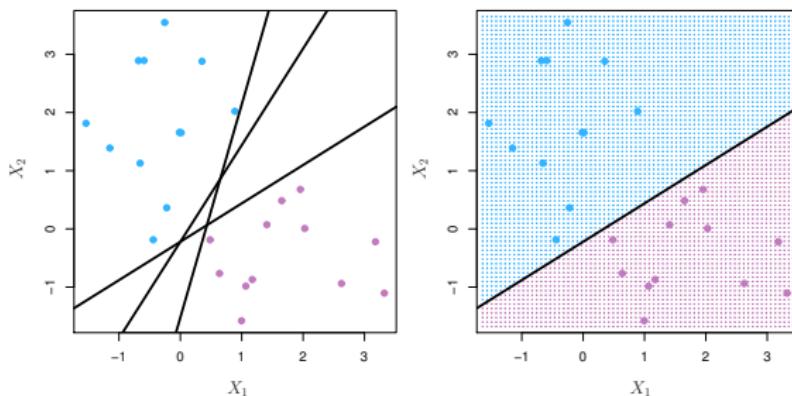
Maximum Margin Idea

- ▶ Suppose we have a classification problem with response $Y = -1$ or $Y = 1$.



Maximum Margin Idea

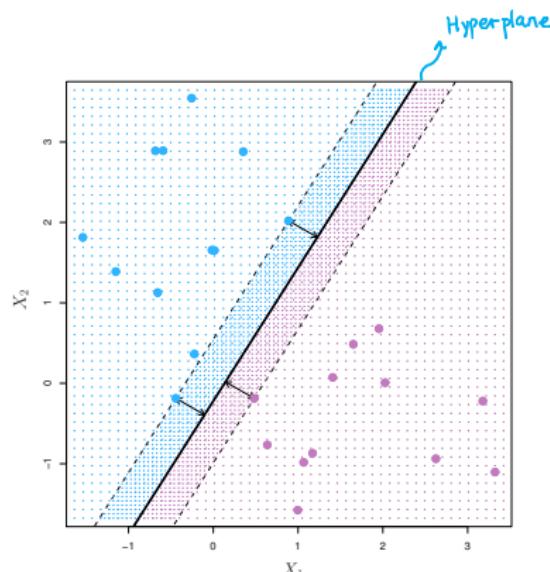
- ▶ Suppose we have a classification problem with response $Y = -1$ or $Y = 1$.
- ▶ If the classes can be separated, most likely, there will be an infinite number of hyperplanes separating the classes.



Maximum Margin Idea

Idea:

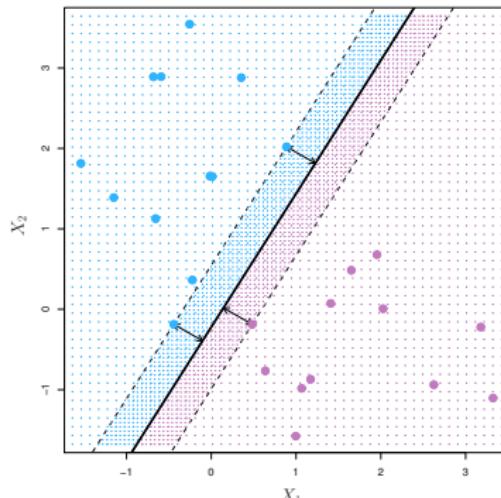
- ▶ Draw the largest possible empty margin around the hyperplane.



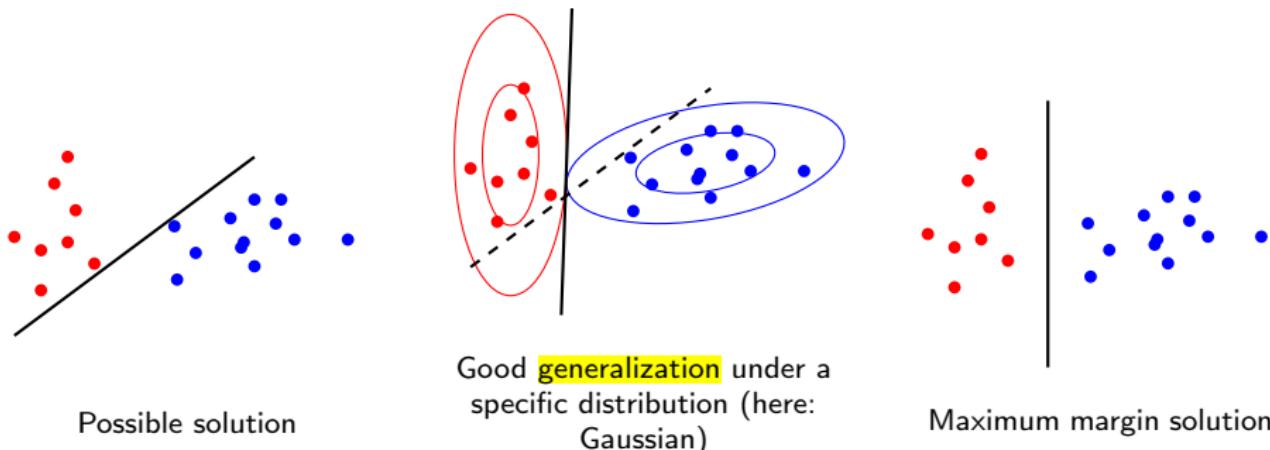
Maximum Margin Idea

Idea:

- ▶ Draw the largest possible empty margin around the hyperplane.
- ▶ Out of all possible hyperplanes that separate the 2 classes, choose the one such that distance to closest point in each class is maximal. This distance is called the *margin*.



Generalization Error

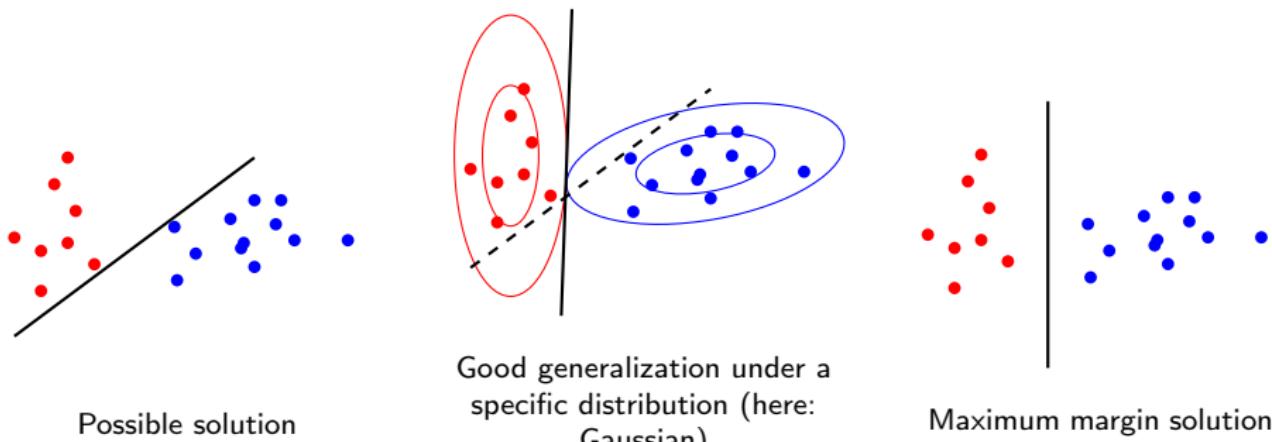


Example: Gaussian data

- ▶ The ellipses represent lines of constant standard deviation (1 and 2 STD respectively).
- ▶ The 1 STD ellipse contains $\sim 65\%$ of the probability mass ($\sim 95\%$ for 2 STD; $\sim 99.7\%$ for 3 STD).

Optimal generalization: Classifier should cut off as little probability mass as possible from either distribution.

Generalization Error



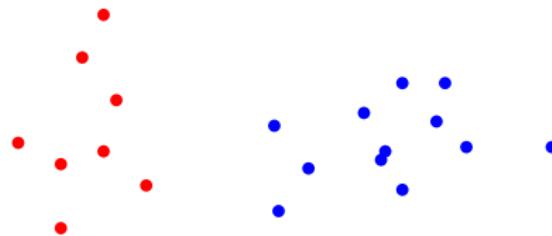
Without distributional assumption: Max-margin classifier

- ▶ Philosophy: Without distribution assumptions, best guess is symmetric.
- ▶ In the Gaussian example, the max-margin solution would *not* be optimal.

Substituting convex sets

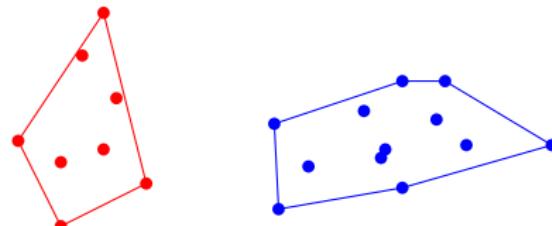
Observation

Where a separating hyperplane may be placed depends on the "outer" points on the sets. Points in the center do not matter.



In geometric terms

Substitute each class by the smallest convex set which contains all point in the class:



Substituting convex sets

Definition

If C is a set of points, the smallest convex set containing all points in C is called the **convex hull** of C , denoted $\text{conv}(C)$.



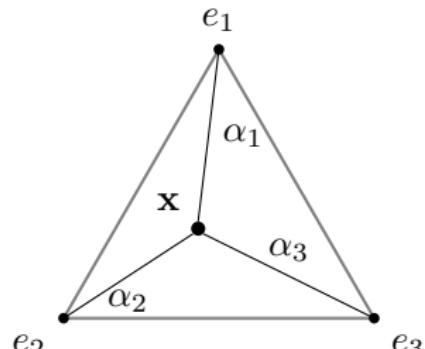
put classifier in the middle of convex hull.

Corner points of the convex set are called **extreme points**.

Barycentric coordinates

Every point x in a convex set can be represented as a **convex combination** of the **extreme points** $\{e_1, \dots, e_m\}$. There are weights $\alpha_1, \dots, \alpha_m \in \mathbb{R}_+$ such that

$$x = \sum_{i=1}^m \alpha_i e_i \quad \text{and} \quad \sum_{i=1}^m \alpha_i = 1.$$

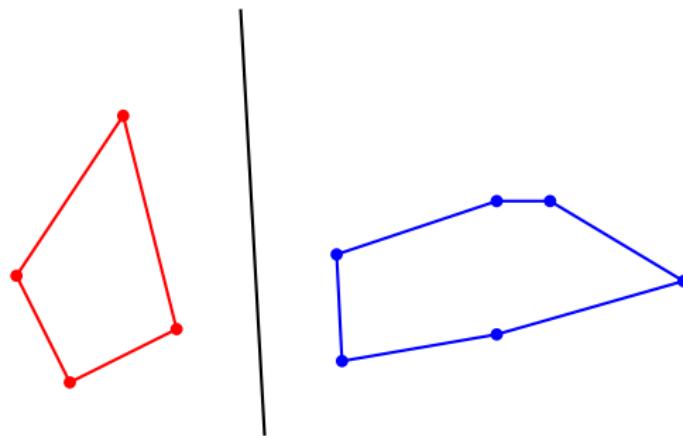


The coefficients α_i are called **barycentric coordinates** of x .

Convex Hulls and Classification

Key idea

A hyperplane separates two classes if and only if it separates their convex hull.



Next: We have to formalize what it means for a hyperplane to be "in the middle" between two classes.

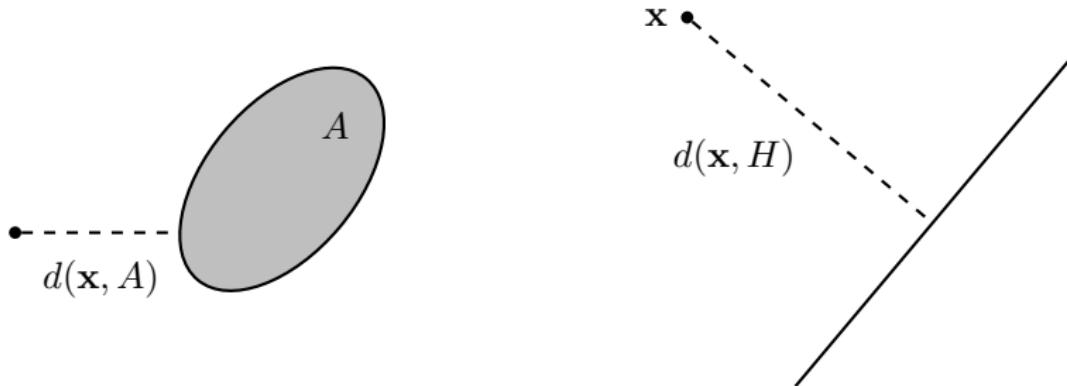
Distances to sets

Definition

The **distance** between a point \mathbf{x} and a set A the Euclidean distance between x and the closest point in A :

$$d(\mathbf{x}, A) := \min_{\mathbf{y} \in A} \|\mathbf{x} - \mathbf{y}\|$$

In particular, if $A = H$ is a hyperplane, $d(\mathbf{x}, H) := \min_{\mathbf{y} \in H} \|\mathbf{x} - \mathbf{y}\|$.



Margin

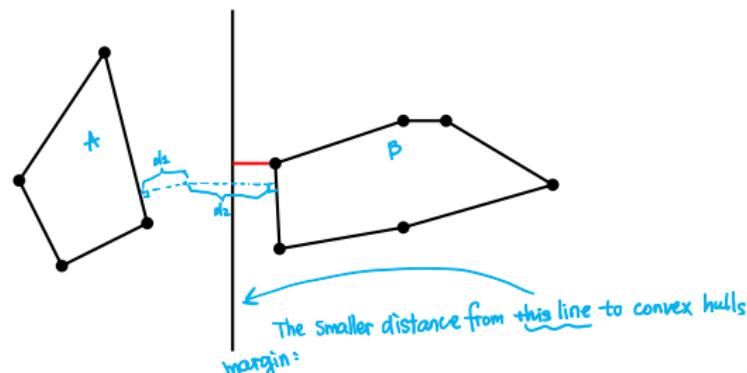
Definition

The **margin** of a classifier hyperplane H given two training classes $\mathcal{X}_{\ominus}, \mathcal{X}_{\oplus}$ is the shortest distance between the plane and any point in either set:

$$\text{margin} = \min_{x \in \mathcal{X}_{\ominus} \cup \mathcal{X}_{\oplus}} d(x, H)$$

Equivalently: The shortest distance to either of the convex hulls.

$$\text{margin} = \min\{d(H, \text{conv}(\mathcal{X}_{\ominus})), d(H, \text{conv}(\mathcal{X}_{\oplus}))\}$$



Idea in the following: H is "in the middle" when margin maximal.

Linear Classifier with Margin

Recall: Specifying affine plane

Normal vector \mathbf{v}_H .

$$\langle \mathbf{v}_H, \mathbf{x} \rangle - c \begin{cases} > 0 & \mathbf{x} \text{ on positive side} \\ < 0 & \mathbf{x} \text{ on negative side} \end{cases}$$

Scalar $c \in \mathbb{R}$ specifies shift (plane through origin if $c = 0$).

Plane with margin

Demand

这一项其实表示的是当 \mathbf{v}_H 的长度为1时 $\langle \mathbf{X} - \mathbf{W}, \mathbf{v}_H \rangle = \cos(\theta) \|\mathbf{X} - \mathbf{W}\|$ 内积值
当 \mathbf{v}_H 的长度为1的时候，其实这内积的结果就是向量到hyperplan的距离，需要这个距离大于1，相当于说强制margin至少为1

$\{-1, 1\}$ on the right works for any margin: Size of margin determined by $\|\mathbf{v}_H\|$. To increase margin, scale down \mathbf{v}_H .

Classification

Concept of margin applies only to training, not to classification.

Classification works as for any linear classifier. For a test point \mathbf{x} :

$$y = \text{sign} (\langle \mathbf{v}_H, \mathbf{x} \rangle - c)$$

Support Vector Machine

Finding the hyperplane don't fix the length of normal vector (\mathbf{v}_H)

For n training points $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ with labels $\tilde{y}_i \in \{-1, 1\}$, solve optimization problem:

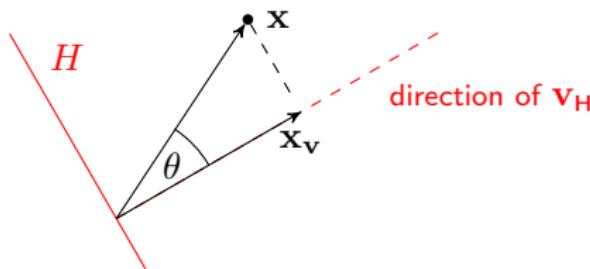
$$\begin{aligned} \min_{\mathbf{v}_H, c} \quad & \|\mathbf{v}_H\| \\ \text{s.t.} \quad & \tilde{y}_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 \quad \text{for } i = 1, \dots, n \\ & \downarrow \\ & \tilde{y}_i \text{ and } (\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \text{ both } > 0 \text{ or both } < 0 \end{aligned}$$

Definition

The classifier obtained by solving this optimization problem is called a **support vector machine**.

Why minimize $\|\mathbf{v}_H\|$?

We can project a vector \mathbf{x} (think: data point) onto the direction of \mathbf{v}_H and obtain a vector \mathbf{x}_v .



- If H has no offset ($c = 0$), the Euclidean distance of \mathbf{x} from H is

$$d(\mathbf{x}, H) = \|\mathbf{x}_v\| = \cos \theta \cdot \|\mathbf{x}\| .$$

It does not depend on the length of \mathbf{v}_H .

- The scalar product $\langle \mathbf{x}, \mathbf{v}_H \rangle$ does increase if the length of \mathbf{v}_H increases.
- To compute the distance $\|\mathbf{x}_v\|$ from $\langle \mathbf{x}, \mathbf{v}_H \rangle$, we have to scale out $\|\mathbf{v}_H\|$:

$$\|\mathbf{x}_v\| = \cos \theta \cdot \|\mathbf{x}\| = \frac{\langle \mathbf{x}, \mathbf{v}_H \rangle}{\|\mathbf{v}_H\|}$$

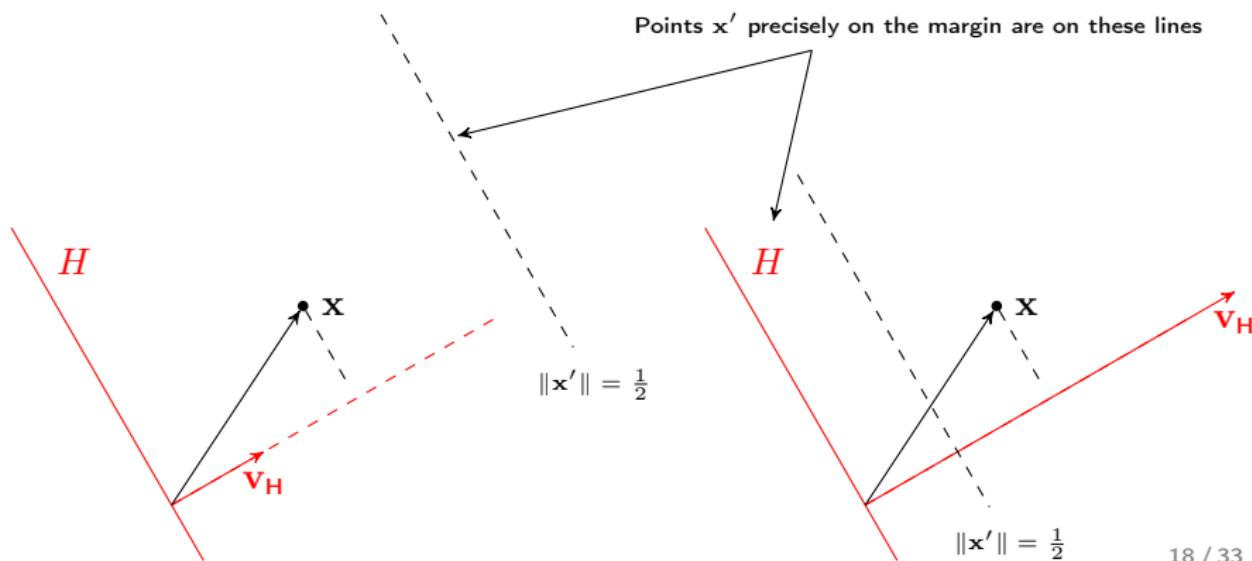
Why minimize $\|\mathbf{v}_H\|$?

If we scale \mathbf{v}_H by α , we have to scale \mathbf{x} by $1/\alpha$ to keep $\langle \mathbf{v}_H, \mathbf{x} \rangle$ constant,
e.g.:

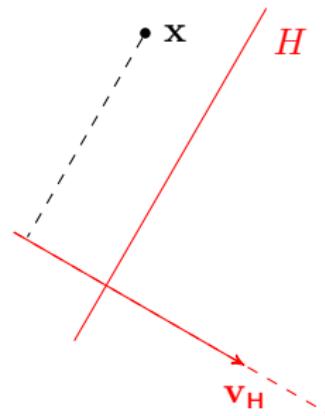
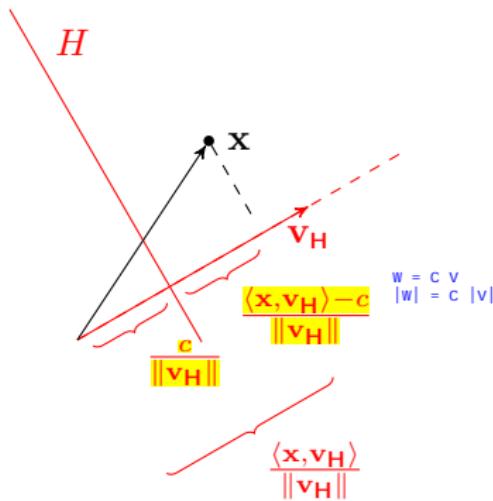
$$1 = \langle \mathbf{v}_H, \mathbf{x} \rangle = \left\langle \alpha \mathbf{v}_H, \frac{1}{\alpha} \mathbf{x} \right\rangle.$$

A point \mathbf{x}' is precisely on the margin if $\langle \mathbf{x}', \mathbf{v}_H \rangle = 1$.

Look at what happens if we scale \mathbf{v}_H :



Distance With Offset



For an affine plane, we have to subtract the offset.

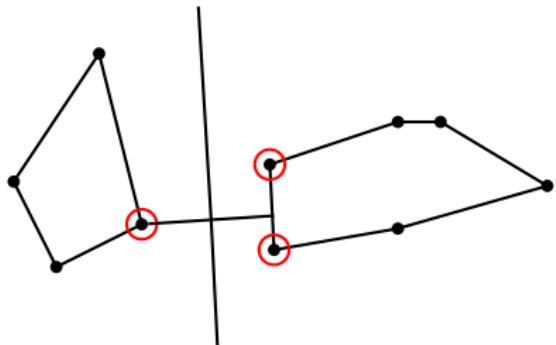
The optimization algorithm can also rotate the vector \mathbf{v}_H , which rotates the plane.

Support Vectors

Definition

Those extreme points of the convex hulls which are closest to the hyperplane are called the **support vectors**.

There are at least two support vectors, one in each class.



Implications

- ▶ The maximum-margin criterion focuses all attention to the area closest to the decision surface.
- ▶ Small changes in the support vectors can result in significant changes of the classifier.
- ▶ In practice, the approach is combined with "slack variables" to permit overlapping classes. As a side effect, slack variables soften the impact of changes in the support vectors.

Dual Optimization Problem

Solving the SVM optimization problem

$$\begin{aligned} \min_{\mathbf{v}_H, c} \quad & \|\mathbf{v}_H\| \\ \text{s.t.} \quad & \tilde{y}_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 \quad \text{for } i = 1, \dots, n \end{aligned}$$

is difficult, because the constraint is a function. It is possible to transform this problem into a problem which seems more complicated, but has simpler constraints:

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & W(\alpha) := \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle \\ \text{s.t.} \quad & \sum_{i=1}^n \tilde{y}_i \alpha_i = 0 \\ & \alpha_i \geq 0 \quad \text{for } i = 1, \dots, n \end{aligned}$$

This is called the optimization problem **dual** to the minimization problem above. It is usually derived using Lagrange multipliers. We will use a more geometric argument.

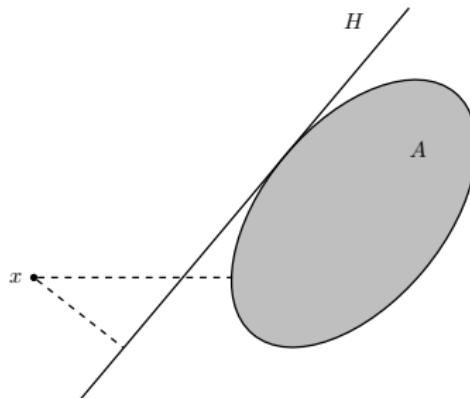
Convex Duality

Sets and Planes

Many dual relations in convex optimization can be traced back to the following fact:

The closest distance between a point \mathbf{x} and a convex set A is the maximum over the distances between \mathbf{x} and all hyperplanes which separate \mathbf{x} and A .

$$d(\mathbf{x}, A) = \sup_{H \text{ separating}} d(\mathbf{x}, H)$$



Deriving the Dual Problem

Idea

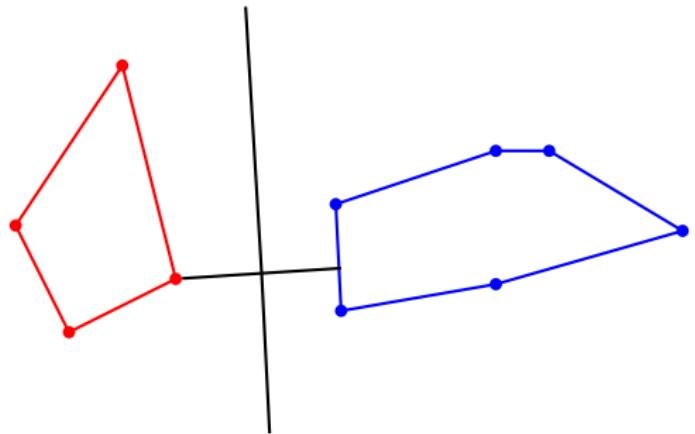
As a consequence of duality on previous slide, we can find the maximum-margin plane as follows:

1. Find shortest line connecting the convex hulls.
2. Place classifier orthogonal to line in the middle.

Convexity of sets ensures that this classifier has correct orientation.

As optimization problem

$$\min_{\mathbf{u} \in \text{conv}(\mathcal{X}_{\ominus}), \mathbf{v} \in \text{conv}(\mathcal{X}_{\oplus})} \|\mathbf{u} - \mathbf{v}\|^2$$



Barycentric Coordinates

Dual optimization problem

$$\min_{\substack{\mathbf{u} \in \text{conv}(\mathcal{X}_{\ominus}) \\ \mathbf{v} \in \text{conv}(\mathcal{X}_{\oplus})}} \|\mathbf{u} - \mathbf{v}\|^2$$

As points in the convex hulls, \mathbf{u} and \mathbf{v} can be represented by barycentric coordinates:

$$\mathbf{u} = \sum_{i=1}^{n_1} \alpha_i \tilde{\mathbf{x}}_i \quad \mathbf{v} = \sum_{i=n_1+1}^{n_1+n_2} \alpha_i \tilde{\mathbf{x}}_i \quad (\text{where } n_1 = |\mathcal{X}_{\ominus}|, n_2 = |\mathcal{X}_{\oplus}|)$$

The extreme points suffice to represent any point in the sets. If $\tilde{\mathbf{x}}_i$ is not an extreme point, we can set $\alpha_i = 0$.

Substitute into minimization problem:

$$\begin{aligned} & \min_{\alpha_1, \dots, \alpha_n} \left\| \sum_{i \in \mathcal{X}_{\ominus}} \alpha_i \tilde{\mathbf{x}}_i - \sum_{i \in \mathcal{X}_{\oplus}} \alpha_i \tilde{\mathbf{x}}_i \right\|_2^2 \\ \text{s.t. } & \sum_{i \in \mathcal{X}_{\ominus}} \alpha_i = \sum_{i \in \mathcal{X}_{\oplus}} \alpha_i = 1, \quad \alpha_i \geq 0 \end{aligned}$$

Dual optimization problem

Dual problem

$$\begin{aligned}\left\| \sum_{i \in \mathcal{X}_{\ominus}} \alpha_i \tilde{\mathbf{x}}_i - \sum_{i \in \mathcal{X}_{\oplus}} \alpha_i \tilde{\mathbf{x}}_i \right\|_2^2 &= \left\| \sum_{i \in \mathcal{X}_{\ominus}} \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i + \sum_{i \in \mathcal{X}_{\oplus}} \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i \right\|_2^2 \\ &= \left\langle \sum_{i=1}^n \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i, \sum_{i=1}^n \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i \right\rangle \\ &= \sum_{i,j} \tilde{y}_i \tilde{y}_j \alpha_i \alpha_j \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle\end{aligned}$$

Note: Minimizing this term under the constraints is equivalent to *maximizing*

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \tilde{y}_i \tilde{y}_j \alpha_i \alpha_j \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle$$

under the same constraints, since $\sum_i \alpha_i = 2$ is constant. That is just the dual problem defined four slides back.

Computing c

Output of dual problem

$$\mathbf{v}_H^* := \mathbf{v}^* - \mathbf{u}^* = \sum_{i=1}^n \tilde{y}_i \alpha_i^* \tilde{\mathbf{x}}_i$$

This vector describes a hyperplane through the origin. We still have to compute the offset.

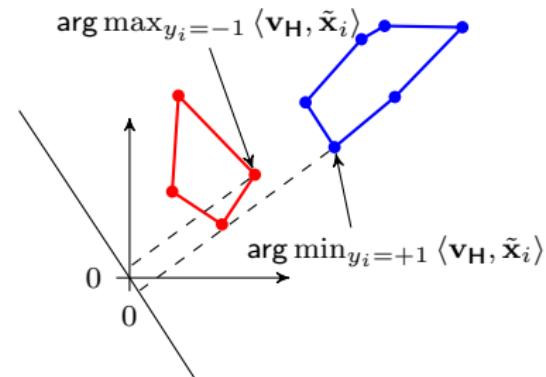
Computing the offset

$$c^* := \frac{\max_{\tilde{y}_i=-1} \langle \mathbf{v}_H^*, \tilde{\mathbf{x}}_i \rangle + \min_{\tilde{y}_i=+1} \langle \mathbf{v}_H^*, \tilde{\mathbf{x}}_i \rangle}{2}$$

Computing c

Explanation

- ▶ The max and min are computed with respect to the \mathbf{v}_H plane *containing the origin*.
- ▶ That means the max and min determine a support vector in each class.
- ▶ We then compute the shift as the mean of the two distances.



Resulting Classification Rule

Output of dual optimization

- ▶ Optimal values α_i^* for the variables α_i
- ▶ If $\tilde{\mathbf{x}}_i$ support vector: $\alpha_i^* > 0$, if not: $\alpha_i^* = 0$

Note: $\alpha_i^* = 0$ holds even if $\tilde{\mathbf{x}}_i$ is an extreme point, but not a support vector.

SVM Classifier

The classification function can be expressed in terms of the variables α_i :

线性分类器

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^n \tilde{y}_i \alpha_i^* \langle \tilde{\mathbf{x}}_i, \mathbf{x} \rangle - c^* \right)$$

Intuitively: To classify a data point, it is sufficient to know which side of each support vector it is on.

Soft-Margin Classifiers

Soft-margin classifiers are maximum-margin classifiers which permit some points to lie on the wrong side of the margin, or even of the hyperplane.

Motivation 1: Nonseparable data

SVMs are linear classifiers; without further modifications, they cannot be trained on a non-separable training data set.

Motivation 2: Robustness

- ▶ Recall: Location of SVM classifier depends on position of (possibly few) support vectors.
- ▶ Suppose we have two training samples (from the same joint distribution on (X, Y)) and train an SVM on each.
- ▶ If locations of support vectors vary significantly between samples, SVM estimate of v_H is “brittle” (depends too much on small variations in training data). → Bad generalization properties.
- ▶ Methods which are not susceptible to small variations in the data are often referred to as **robust**.

Slack Variables

Idea

Permit training data to cross the margin, but impose cost which increases the further beyond the margin we are.

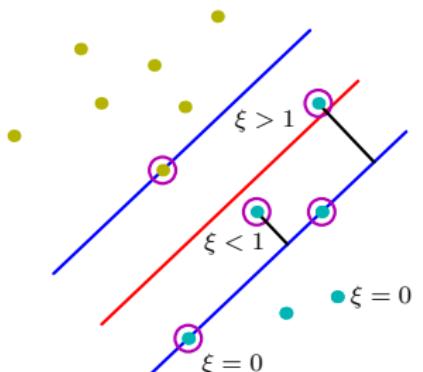
Formalization

We replace the training rule $\tilde{y}_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \geq 1$ by

$$\tilde{y}_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 - \xi_i$$

with $\xi_i \geq 0$. The variables ξ_i are called **slack variables**.

当slack variable大于1的时候就会出现允许部分数据可以出现在错误的边界的情况



Soft-Margin SVM

Soft-margin optimization problem

$$\begin{aligned} \min_{\mathbf{v}_H, c, \xi} \quad & \|\mathbf{v}_H\|^2 + \gamma \sum_{i=1}^n \xi_i^2 \\ \text{s.t.} \quad & \tilde{y}_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 - \xi_i \quad \text{for } i = 1, \dots, n \\ & \xi_i \geq 0, \quad \text{for } i = 1, \dots, n \end{aligned}$$

parameter 在这里相当于 slack variable 的惩罚项，如果 parameter 很小，那么每一个 slack variable 就可以允许很大。相反，如果我们取一个很大的 parameter，那么 slack variable 就一定要比较小才可能实现最小化

The training algorithm now has a **parameter** $\gamma > 0$ for which we have to choose a "good" value. γ is usually set by *cross validation* (discussed later). Its value is fixed before we start the optimization.

Role of γ

- ▶ Specifies the "cost" of allowing a point on the wrong side.
- ▶ If γ is very small, many points may end up beyond the margin boundary.
- ▶ For $\gamma \rightarrow \infty$, we recover the original SVM.

Soft-Margin SVM

Soft-margin dual problem

The slack variables vanish in the dual problem.

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & W(\alpha) := \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j (\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle + \frac{1}{\gamma} \mathbb{I}\{i = j\}) \\ \text{s.t.} \quad & \sum_{i=1}^n \tilde{y}_i \alpha_i = 0 \\ & \alpha_i \geq 0 \quad \text{for } i = 1, \dots, n \end{aligned}$$

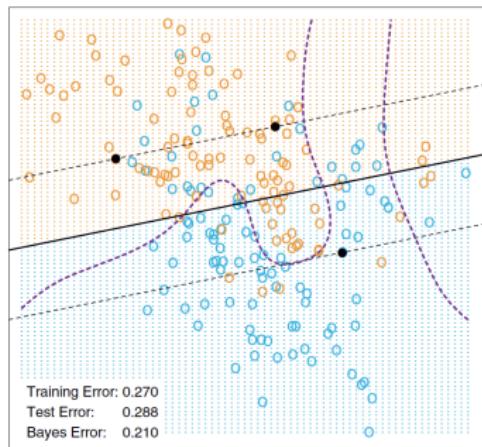
Soft-margin classifier

The classifier looks exactly as for the original SVM:

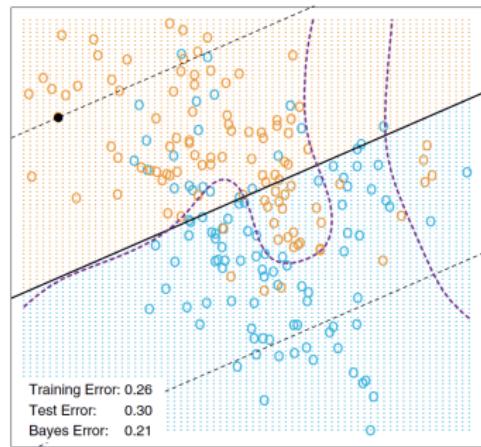
$$f(\mathbf{x}) = \operatorname{sgn} \left(\sum_{i=1}^n \tilde{y}_i \alpha_i^* \langle \tilde{\mathbf{x}}_i, \mathbf{x} \rangle - c \right)$$

Note: Each point on wrong side of the margin is an additional support vector ($\alpha_i^* \neq 0$), so the ratio of support vectors can be substantial when classes overlap.

Influence of Margin Parameter



$$\gamma = 100000$$



$$\gamma = 0.01$$

Changing γ significantly changes the classifier (note how the slope changes in the figures). We need a method to select an appropriate value of γ , in other words: to learn γ from data.

for convex optimization: the local minimum should also be global minimum

Lecture 8: Introduction to Convex Optimization

EE 364 AB

Reading: *Convex optimization* by Boyd and Vandenberghe.
package CVX

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
February 16, 2018

Optimization Problems

Terminology

An **optimization problem** for a given function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is a problem of the form

$$\min_{\mathbf{x}} f(\mathbf{x})$$

which we read as "find $\mathbf{x}_0 = \arg \min_{\mathbf{x}} f(\mathbf{x})$ ".

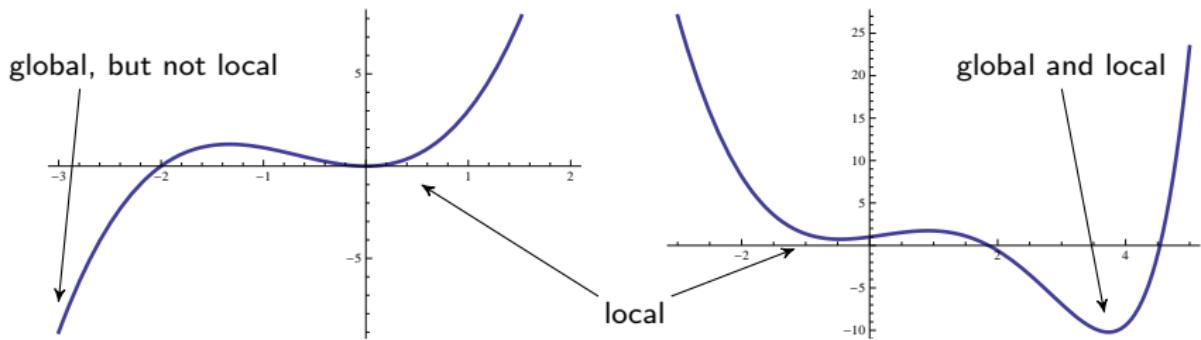
A **constrained optimization problem** adds additional requirements on \mathbf{x} ,

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{subject to } & \mathbf{x} \in G, \end{aligned}$$

where $\text{dom} f \cap G \subset \mathbb{R}^d$ is called the **feasible set**. The set G is often defined by equations, e.g.

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{subject to } & g(\mathbf{x}) \geq 0 \end{aligned}$$

Types of Minima



Local and global minima

A minimum of f at x is called:

- ▶ **Global** if f assumes no smaller value on its domain.
- ▶ **Local** if there is some open neighborhood U of x such that $f(x)$ is a global minimum of f restricted to U .

Optima

Analytic criteria for local minima

Recall that \mathbf{x} is a local minimum of f if

严格的local 最小值点
 $f'(\mathbf{x}) = 0 \quad \text{and} \quad f''(\mathbf{x}) > 0 .$

In \mathbb{R}^d ,

$$\nabla f(\mathbf{x}) = 0 \quad \text{and} \quad H_f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_i \partial x_j}(\mathbf{x}) \right)_{i,j=1,\dots,n} \text{ positive definite.}$$

The $d \times d$ -matrix $H_f(\mathbf{x})$ is called the **Hessian matrix** of f at \mathbf{x} .

Optima

1. un-constrain problem

$\min f(x)$

2. constrain problem

$\min f(x)$

s.t. $g(x) = 0$

or s.t. $g(x) < 0$

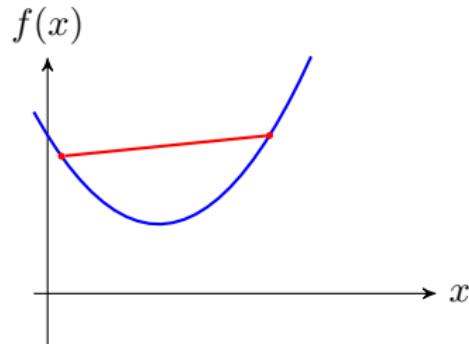
Numerical methods

All numerical minimization methods perform roughly the same steps:

- ▶ Start with some point x_0 .
- ▶ Our goal is to find a sequence x_0, \dots, x_m such that $f(x_m)$ is a minimum.
- ▶ At a given point x_n , compute properties of f (such as $f'(x_n)$ and $f''(x_n)$).
- ▶ Based on these values, choose the next point x_{n+1} .

The information $f'(x_n)$, $f''(x_n)$ etc is always *local at x_n* , and we can only decide whether a point is a local minimum, not whether it is global.

Convex Functions



Definition

A function f is **convex** if every line segment between function values lies above the graph of f .

Analytic criterion

A twice differentiable function is convex if $f''(x) \geq 0$ (or $H_f(\mathbf{x})$ positive semidefinite) for all \mathbf{x} .

Implications for optimization

If f is convex, then:

two requirement for the convex problem
1. objective function is a convex function
2. set is a convex set
(for any given tow point in the set, if we draw the line, the line should be strictly inside the set)

- ▶ $f'(x) = 0$ is a sufficient criterion for a minimum.
- ▶ Local minima are global.
- ▶ If f is **strictly convex** ($f'' > 0$ or H_f positive definite), there is only one minimum (which is both global and local).

Gradient Descent

first order method

Algorithm

gradient will tell the direction where the value changes most importantly

Gradient descent searches for a **minimum of f** .

1. Start with some point $x \in \mathbb{R}$ and fix a precision $\varepsilon > 0$.
2. Repeat for $n = 1, 2, \dots$

fastest ascent: the positive
fastest descent: the negative

$$x_{n+1} := x_n - f'(x_n)$$

3. Terminate when $|f'(x_n)| < \varepsilon$.

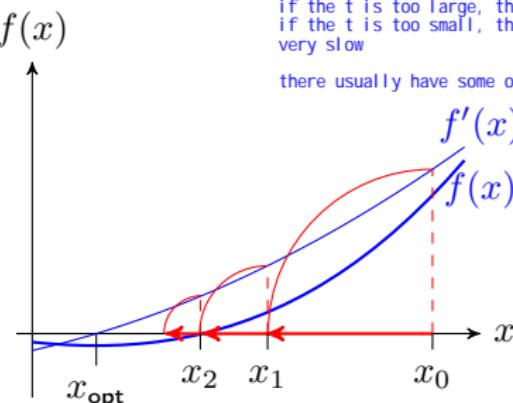
step set here is just 1

we can set the step set which is the learning rate

$$x_{n+1} := x_n - t * f'(x_n)$$

if the t is too large, this will end that we can not find the minimum
if the t is too small, this will end that we will find the minimum very slow

there usually have some optimum value of t



Newton's Method: Roots

Algorithm

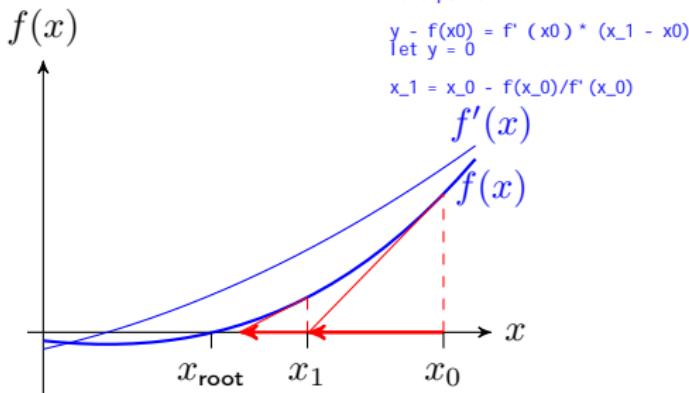
Newton's method searches for a **root** of f , i.e. it solves the equation $f(\mathbf{x}) = 0$.

1. Start with some point $x \in \mathbb{R}$ and fix a precision $\varepsilon > 0$.
2. Repeat for $n = 1, 2, \dots$

$$x_{n+1} := x_n - f(x_n)/f'(x_n)$$

3. Terminate when $|f(x_n)| < \varepsilon$.

the biggest difference here we just draw a line to find the next point.



Basic Applications

Function evaluation

Most numerical evaluations of functions (\sqrt{a} , $\sin(a)$, $\exp(a)$, etc) are implemented using Newton's method. To evaluate g at a , we have to transform $x = g(a)$ into an equivalent equation of the form

$$f(x, a) = 0 .$$

We then fix a and solve for x using Newton's method for roots.

Example: Square root

To evaluate $g(a) = \sqrt{a}$, we can solve

$$f(x, a) = x^2 - a = 0 .$$

This is essentially how `sqrt()` is implemented in the standard C library.

Newton's Method: Minima

usually for the newton's method
it will take less step to converge

however we also need to consider the
computational cost

newton's method the computational cost is
every high

thus in the high dimensional case, the
newton's method would cost more time.

Algorithm

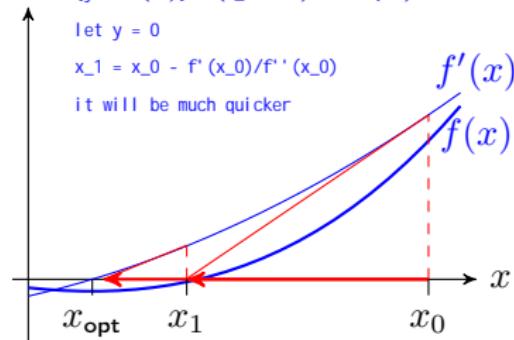
We can use Newton's method for minimization by applying it to solve
 $f'(x) = 0$.

1. Start with some point $x \in \mathbb{R}$ and fix a precision $\varepsilon > 0$.
2. Repeat for $n = 1, 2, \dots$

$$x_{n+1} := x_n - f'(x_n)/f''(x_n)$$

3. Terminate when $|f'(x_n)| < \varepsilon$.

the biggest difference here we just draw a line to find the next point.
First we calculate the $f'(x)$ function , we just need to find the point which enable the
function to be zero
 $(y - f'(x_0)) / (x_{-1} - x_0) = f''(x_0)$



Multiple Dimensions

In \mathbb{R}^d we have to replace the derivatives by their vector space analogues.

Gradient descent

$$\mathbf{x}_{n+1} := \mathbf{x}_n - \nabla f(\mathbf{x}_n)$$

Newton's method for minima

$$\mathbf{x}_{n+1} := \mathbf{x}_n - H_f^{-1}(\mathbf{x}_n) \cdot \nabla f(\mathbf{x}_n)$$

The inverse of $H_f(\mathbf{x})$ exists only if the matrix is positive definite (not if it is only semidefinite), i.e. f has to be strictly convex.

The Hessian measures the curvature of f .

Effect of the Hessian

Multiplication by H_f^{-1} in general changes the direction of $\nabla f(\mathbf{x}_n)$. The correction takes into account how $\nabla f(\mathbf{x})$ changes away from \mathbf{x}_n , as estimated using the Hessian at \mathbf{x}_n .

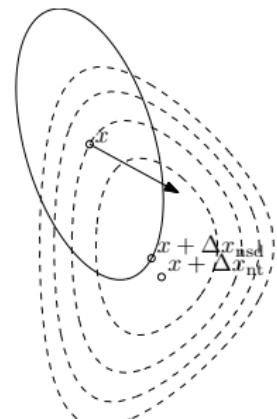


Figure: Arrow is ∇f , $x + \Delta x_{\text{nt}}$ is Newton step.

Newton: Properties

Convergence

- ▶ The algorithm always converges if $f'' > 0$ (or H_f positive definite).
- ▶ The speed of convergence separates into two phases:
 - ▶ In a (possibly small) region around the minimum, f can always be approximated by a quadratic function.
 - ▶ Once the algorithm reaches that region, the error decreases at quadratic rate. Roughly speaking, the number of correct digits in the solution doubles in each step.
 - ▶ Before it reaches that region, the convergence rate is linear.

High dimensions

- ▶ The required number of steps hardly depends on the dimension of \mathbb{R}^d . Even in \mathbb{R}^{10000} , you can usually expect the algorithm to reach high precision in half a dozen steps.
- ▶ Caveat: The individual steps can become very expensive, since we have to invert H_f in each step, which is of size $d \times d$.

Next: Constrained Optimization

So far

- ▶ If f is differentiable, we can search for local minima using gradient descent.
- ▶ If f is sufficiently nice (convex and twice differentiable), we know how to speed up the search process using Newton's method.

Constrained problems

- ▶ The numerical minimizers use the criterion $\nabla f(x) = 0$ for the minimum.
- ▶ In a constrained problem, the minimum is *not* identified by this criterion.

Next steps

We will figure out how the constrained minimum can be identified. We have to distinguish two cases:

- ▶ Problems involving only equalities as constraints (easy).
- ▶ Problems also involving inequalities (a bit more complex).

Optimization Under Constraints

Objective

equality constraints

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{subject to } g(\mathbf{x}) = 0 \end{aligned}$$

Idea

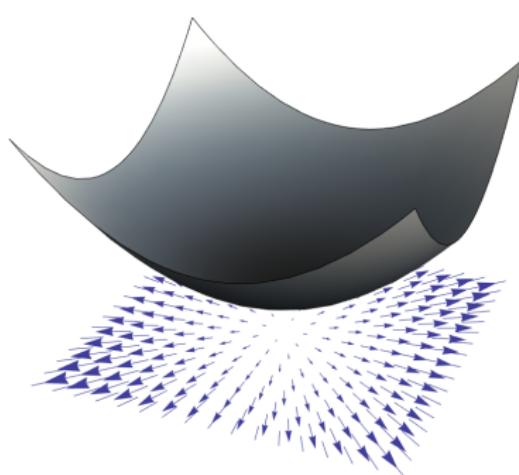
- The feasible set is the set of points \mathbf{x} which satisfy $g(\mathbf{x}) = 0$,

$$G := \{\mathbf{x} \mid g(\mathbf{x}) = 0\}.$$

If g is reasonably smooth, G is a smooth surface in \mathbb{R}^d .

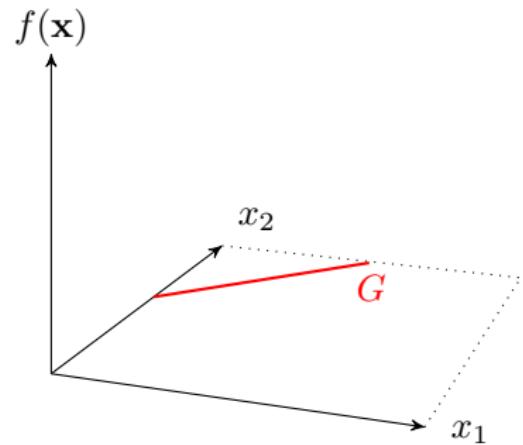
- We restrict the function f to this surface and call the restricted function f_g .
- The constrained optimization problem says that we are looking for the minimum of f_g .

Lagrange Optimization



$$f(\mathbf{x}) = x_1^2 + x_2^2$$

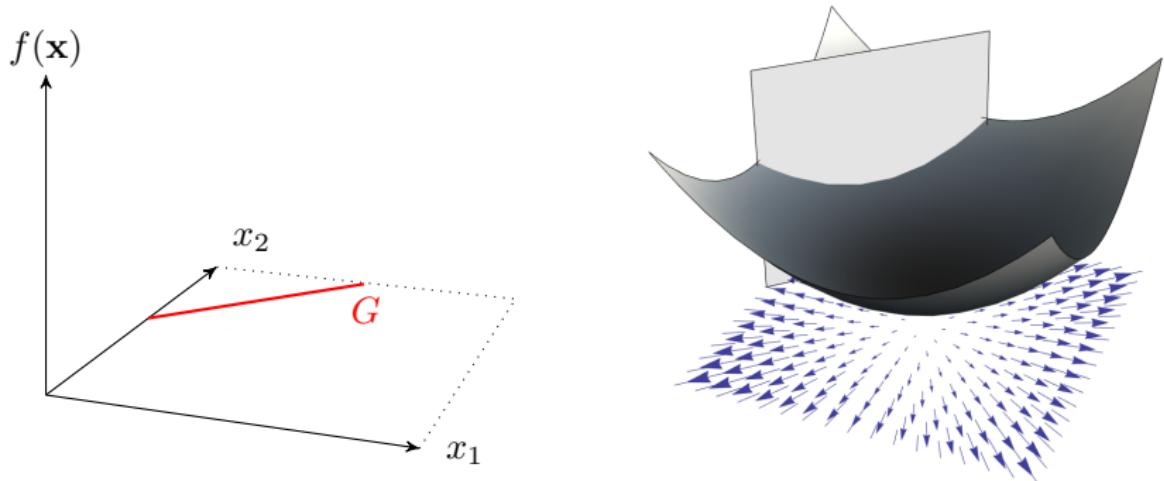
The blue arrows are the gradients $\nabla f(\mathbf{x})$ at various values of \mathbf{x} .



Constraint g .

Here, g is linear, so the graph of g is a (sloped) affine plane. The intersection of the plane with the x_1 - x_2 -plane is the set G of all points \mathbf{x} with $g(\mathbf{x}) = 0$.

Lagrange Optimization



- We can make the function f_g given by the constraint $g(\mathbf{x}) = 0$ visible by placing a plane vertically through G . The graph of f_g is the intersection of the graph of f with the plane.
- Here, f_g has parabolic shape.
- The gradient of f at the minimum of f_g is *not* 0.

Gradients and Contours

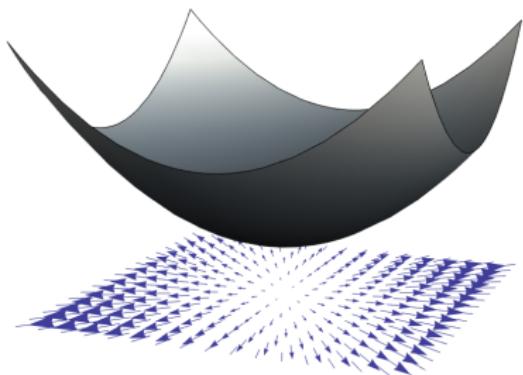
Fact

Gradients are orthogonal to contour lines.

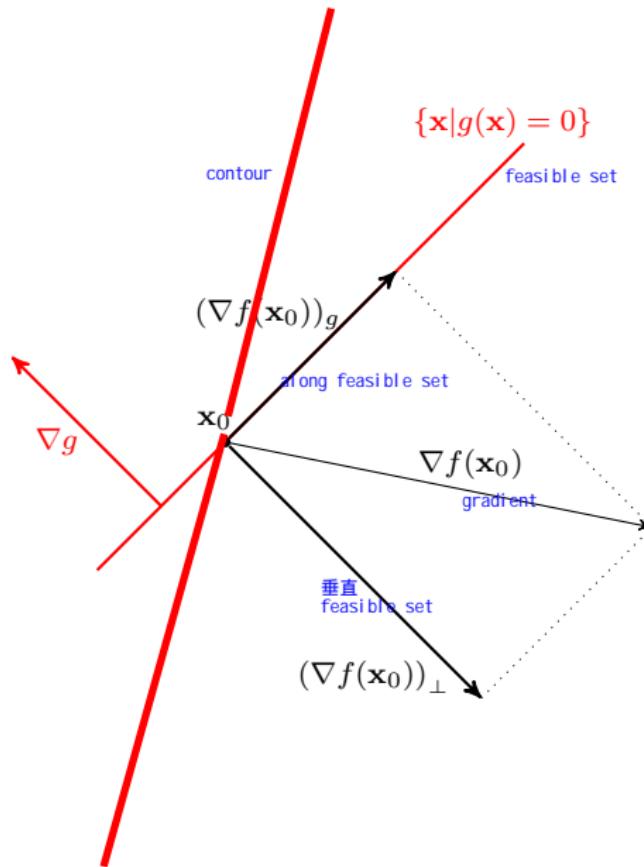
contours is the place there the value will remain the same

Intuition

- ▶ The gradient points in the direction in which f grows most rapidly.
- ▶ Contour lines are sets along which f does not change.



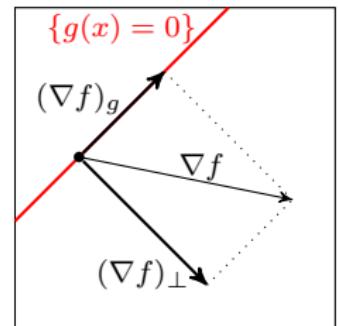
The Crucial Bit



Again, in detail.

Idea

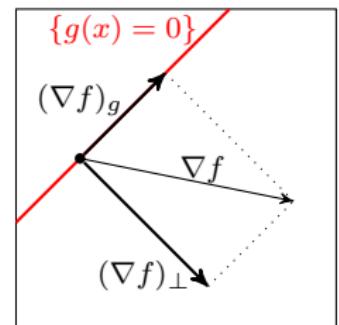
- ▶ Decompose ∇f into a component $(\nabla f)_g$ in the set $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ and a remainder $(\nabla f)_{\perp}$.
- ▶ The two components are orthogonal.
- ▶ If f_g is minimal within $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$, the component within the set vanishes.
- ▶ The remainder need not vanish.



Again, in detail.

Idea

- ▶ Decompose ∇f into a component $(\nabla f)_g$ in the set $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ and a remainder $(\nabla f)_{\perp}$.
- ▶ The two components are orthogonal.
- ▶ If f_g is minimal within $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$, the component within the set vanishes.
- ▶ The remainder need not vanish.



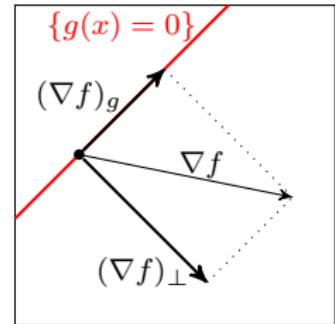
Consequence

- ▶ We need a criterion for $(\nabla f)_g = 0$.

Again, in detail.

Idea

- ▶ Decompose ∇f into a component $(\nabla f)_g$ in the set $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ and a remainder $(\nabla f)_{\perp}$.
- ▶ The two components are orthogonal.
- ▶ If f_g is minimal within $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$, the component within the set vanishes.
- ▶ The remainder need not vanish.



Solution

- ▶ If $(\nabla f)_g = 0$, then ∇f is orthogonal to the set $g(\mathbf{x}) = 0$.
- ▶ Since gradients are orthogonal to contours, and the set is a contour of g , ∇g is also orthogonal to the set.
- ▶ Hence: At a minimum of f_g , the two gradients point in the same direction: $\nabla f + \lambda \nabla g = 0$ for some scalar $\lambda \neq 0$.

Solution: Constrained Optimization

Solution

The constrained optimization problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) = 0 \end{aligned}$$

is solved by solving the equation system

$$\nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0$$

有一个限制条件就引入一个拉格朗日乘子
然后分别对不同的自变量求偏导数
因此这里的第一个式子其实展开是D个式子

$$g(\mathbf{x}) = 0$$

The vectors ∇f and ∇g are D -dimensional, so the system contains $D + 1$ equations for the $D + 1$ variables x_1, \dots, x_D, λ .

Inequality Constraints

Objective

For a function f and a convex function g , solve

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{subject to } g(\mathbf{x}) \leq 0 \end{aligned}$$

i.e. we replace $g(\mathbf{x}) = 0$ as previously by $g(\mathbf{x}) \leq 0$. This problem is called an optimization problem with **inequality constraint**.

Feasible set

We again write G for the set of all points which satisfy the constraint,

$$G := \{\mathbf{x} \mid g(\mathbf{x}) \leq 0\}.$$

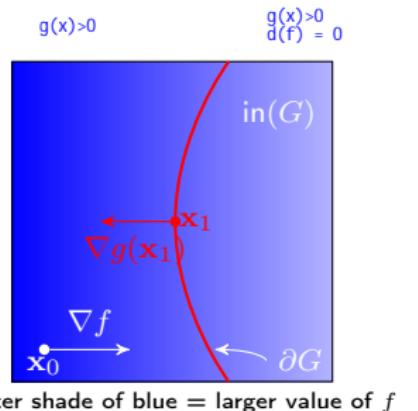
G is often called the **feasible set** (the same name is used for equality constraints).

Two Cases

Case distinction

1. The location x of the minimum can be in the *interior* of G the direction of the gradient should be opposite otherwise there will not have a minimum on the boundary
2. x may be on the *boundary* of G .

Decomposition of G



$$G = \text{in}(G) \cup \partial G = \text{interior} \cup \text{boundary}$$

Note: The interior is given by $g(\mathbf{x}) < 0$, the boundary by $g(\mathbf{x}) = 0$.

Criteria for minimum

1. **In interior:** $f_g = f$ and hence $\nabla f_g = \nabla f$. We have to solve a standard optimization problem with criterion $\nabla f = 0$.
2. **On boundary:** Here, $\nabla f_g \neq \nabla f$. Since $g(\mathbf{x}) = 0$, the geometry of the problem is the same as we have discussed for equality constraints, with criterion $\nabla f = \lambda \nabla g$.
However: In this case, the sign of λ matters.

On the Boundary

Observation

- ▶ An extremum on the boundary is a minimum only if ∇f points *into* G .
- ▶ Otherwise, it is a maximum instead.

Criterion for minimum on boundary

Since ∇g points away from G (since g increases away from G), ∇f and ∇g have to point in opposite directions:

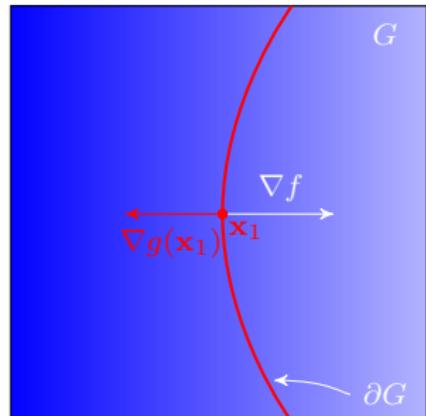
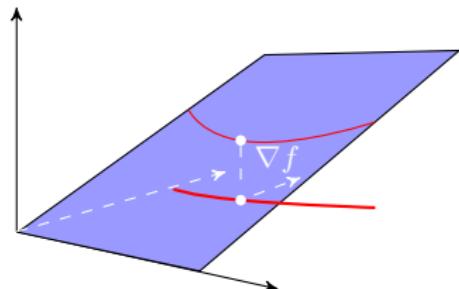
$$\nabla f = \lambda \nabla g \quad \text{with } \lambda \leq 0$$

Convention

To make the sign of λ explicit, we constrain λ to positive values and instead write:

$$\nabla f = -\lambda \nabla g$$

$$\text{s.t. } \lambda \geq 0$$



Combining the Cases

Combined problem

$$\begin{aligned}\nabla f &= -\lambda \nabla g \\ \text{s.t.} \quad g(\mathbf{x}) &\leq 0 \\ \lambda &= 0 \text{ if } \mathbf{x} \in \text{in}(G) \\ \lambda &> 0 \text{ if } \mathbf{x} \in \partial G\end{aligned}$$

Can we get rid of the "if $\mathbf{x} \in \cdot$ " distinction?

Yes: Note that $g(\mathbf{x}) < 0$ if \mathbf{x} in interior and $g(\mathbf{x}) = 0$ on boundary.
Hence, we always have either $\lambda = 0$ or $g(\mathbf{x}) = 0$ (and never both).

That means we can substitute

$$\begin{aligned}\lambda &= 0 \text{ if } \mathbf{x} \in \text{in}(G) \\ \lambda &> 0 \text{ if } \mathbf{x} \in \partial G\end{aligned}$$

by

$$\lambda \cdot g(\mathbf{x}) = 0 \quad \text{and} \quad \lambda \geq 0 .$$

Solution: Inequality Constraints

Combined solution

The optimization problem with inequality constraints

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{subject to } g(\mathbf{x}) \leq 0 \end{aligned}$$

can be solved by solving

s.t.

$$\left. \begin{array}{l} \nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x}) \\ \lambda g(\mathbf{x}) = 0 \\ g(\mathbf{x}) \leq 0 \\ \lambda \geq 0 \end{array} \right\} \xleftarrow{\text{complimentary select}} \text{system of } d+1 \text{ equations for } d+1 \text{ variables } x_1, \dots, x_D, \lambda$$

These conditions are known as the **Karush-Kuhn-Tucker** (or KKT) conditions.

Remarks

Haven't we made the problem more difficult?

- ▶ To simplify the minimization of f for $g(\mathbf{x}) \leq 0$, we have made f more complicated and added a variable and two constraints. Well done.
- ▶ However: In the original problem, we *do not know how to minimize* f , since the usual criterion $\nabla f = 0$ does not work.
- ▶ By adding λ and additional constraints, we have reduced the problem to solving a system of equations.

Summary: Conditions

Condition	Ensures that...	Purpose
$\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$	If $\lambda = 0$: ∇f is 0 If $\lambda > 0$: ∇f is anti-parallel to ∇g	Opt. criterion inside G Opt. criterion on boundary
$\lambda g(\mathbf{x}) = 0$	$\lambda = 0$ in interior of G	Distinguish cases in(G) and ∂G
$\lambda \geq 0$	∇f cannot flip to orientation of ∇g	Optimum on ∂G is minimum

Why Should g be Convex?

More precisely

If g is a convex function, then

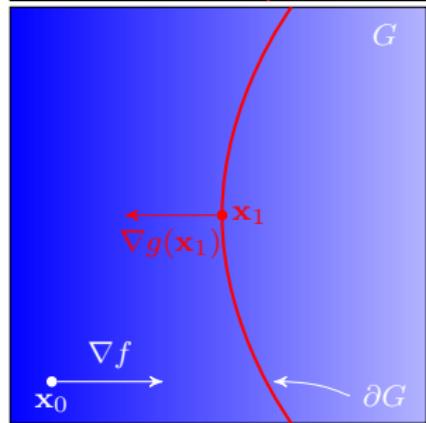
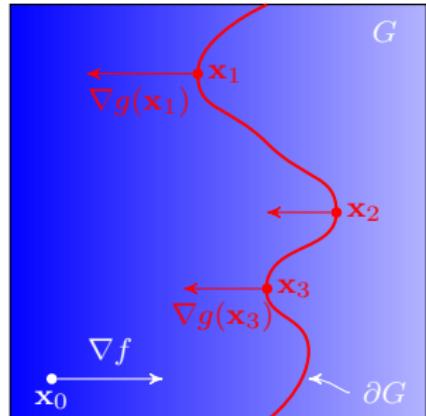
$G = \{\mathbf{x} \mid g(\mathbf{x}) \leq 0\}$ is a convex set. Why do we require convexity of G ?

Problem

If G is not convex, the KKT conditions do not guarantee that \mathbf{x} is a minimum. (The conditions still hold, i.e. if G is not convex, they are necessary conditions, but not sufficient.)

Example (Figure)

- ▶ f is a linear function (lighter color = larger value)
- ▶ ∇f is identical everywhere
- ▶ If G is not convex, there can be several points ($\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$) which satisfy the KKT conditions. Only \mathbf{x}_1 minimizes f on G .
- ▶ G is convex, such problems cannot occur.



Interior Point Methods

Numerical methods for constrained problems

Once we have transformed our problem using Lagrange multipliers, we still have to solve a problem of the form

$$\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$$

$$\text{s.t.} \quad \lambda g(\mathbf{x}) = 0 \quad \text{and} \quad g(\mathbf{x}) \leq 0 \quad \text{and} \quad \lambda \geq 0$$

numerically.

Barrier functions

Idea

A constraint in the problem

$$\min f(x) \quad \text{s.t.} \quad g(x) < 0$$

can be expressed as an indicator function:

$$\min f(x) + \text{const.} \cdot \mathbb{I}_{[0,\infty)}(g(x))$$

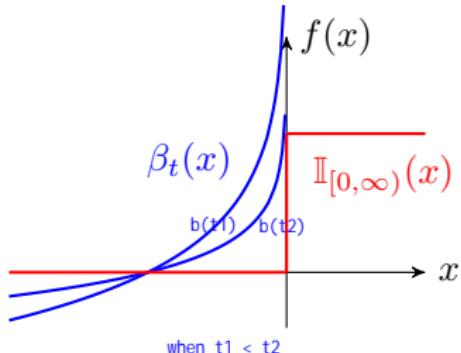
The constant must be chosen large enough to enforce the constraint.

Problem: The indicator function is piece-wise constant and not differentiable at 0. Newton or gradient descent are not applicable.

Barrier function

A **barrier function** approximates $\mathbb{I}_{[0,\infty)}$ by a smooth function, e.g.

$$\beta_t(x) := -\frac{1}{t} \log(-x) .$$



Newton for Constrained Problems

Interior point methods

We can (approximately) solve

$$\min f(x) \text{ s.t. } g_i(x) < 0 \quad \text{for } i = 1, \dots, m$$

by solving

$$\min f(x) + \sum_{i=1}^m \beta_{i,t}(x) .$$

with one barrier function $\beta_{i,t}$ for each constraint g_i .

We do not have to adjust a multiplicative constant since $\beta_t(x) \rightarrow \infty$ as $x \nearrow 0$.

Constrained problems: General solution strategy

1. Convert constraints into solvable problem using Lagrange multipliers.
2. Convert constraints of transformed problem into barrier functions.
3. Apply numerical optimization (usually Newton's method).

Recall: SVM

Original optimization problem

$$\min_{\mathbf{v}_H, c} \|\mathbf{v}_H\|_2 \quad \text{s.t.} \quad y_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 \quad \text{for } i = 1, \dots, n$$

Problem with inequality constraints $g_i(\mathbf{v}_H) \leq 0$ for
 $g_i(\mathbf{v}_H) := 1 - y_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c)$.

Transformed problem

If we transform the problem using Lagrange multipliers $\alpha_1, \dots, \alpha_n$, we obtain:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad W(\boldsymbol{\alpha}) := \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle$$

$$\text{s.t.} \quad \sum_{i=1}^n \tilde{y}_i \alpha_i = 0$$

$$\alpha_i \geq 0 \quad \text{for } i = 1, \dots, n$$

This is precisely the "dual problem" we obtained before using geometric arguments. We can find the max-margin hyperplane using an interior point method.

Relevance in Statistics

Minimization problems

Most methods that we encounter in this class can be phrased as minimization problem. For example:

Problem	Objective function
ML estimation	negative log-likelihood
Classification	empirical risk
Regression	fitting or prediction error
Unsupervised learning	suitable cost function (later)

More generally

The lion's share of algorithms in statistics or machine learning fall into either of two classes:

1. Optimization methods.
2. Simulation methods (e.g. Markov chain Monte Carlo algorithms).

Lecture 9: Support Vector Machines II

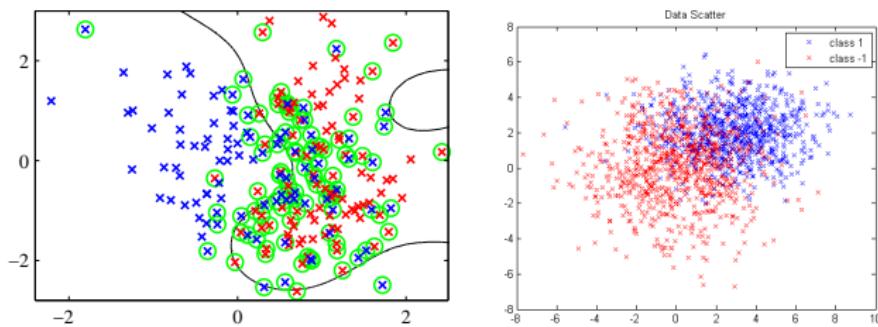
Reading: Section 12.3

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
February 16, 2018

Motivation

More realistic data



Motivation: Kernels

Idea

- ▶ The SVM uses the scalar product $\langle \mathbf{x}, \tilde{\mathbf{x}}_i \rangle$ as a measure of similarity between \mathbf{x} and $\tilde{\mathbf{x}}_i$, and of distance to the hyperplane.
- ▶ Since the scalar product is linear, the SVM is a linear method.
- ▶ By using a *nonlinear* function instead, we can make the classifier nonlinear.

Kernels in Detail

- ▶ Scalar product can be regarded as a two-argument function

$$\langle \cdot, \cdot \rangle : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$$

- ▶ We will replace this function with a function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ and substitute

$$k(\mathbf{x}, \mathbf{x}') \quad \text{for every occurrence of} \quad \langle \mathbf{x}, \mathbf{x}' \rangle$$

in the SVM formula.

- ▶ Under certain conditions on k , all optimization/classification results for the SVM still hold. Functions that satisfy these conditions are called **kernel functions**.

The Most Popular Kernel

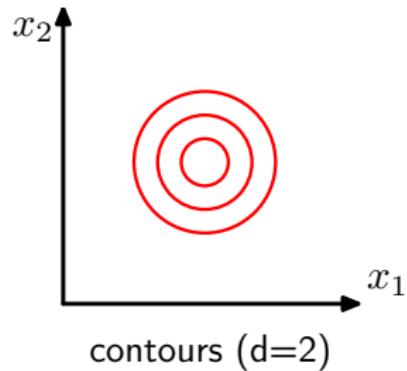
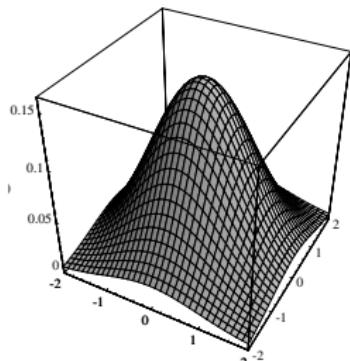
RBF Kernel

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{x}') := \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\sigma^2}\right) \quad \text{for some } \sigma \in \mathbb{R}_+$$

is called an **RBF kernel** (RBF = radial basis function). The parameter σ is called **bandwidth**.

Other names for k_{RBF} : Gaussian kernel, squared-exponential kernel.

If we fix \mathbf{x}' , the function $k_{\text{RBF}}(., \mathbf{x}')$ is (up to scaling) a spherical Gaussian density on \mathbb{R}^d , with mean \mathbf{x}' and standard deviation σ .



Choosing a kernel

Theory

To define a kernel:

- ▶ We have to define a function of two arguments and prove that it is a kernel.
- ▶ This is done by checking a set of necessary and sufficient conditions known as “Mercer’s theorem”.

Practice

The data analyst does not define a kernel, but tries some well-known standard kernels until one seems to work. Most common choices:

- ▶ The RBF kernel.
- ▶ The "linear kernel" $k_{SP}(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$, i.e. the standard, linear SVM.

Once kernel is chosen

- ▶ Classifier can be trained by solving the optimization problem using standard software.
- ▶ SVM software packages include implementations of most common kernels.

Which Functions work as Kernels?

Formal definition

A function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is called a **kernel** on \mathbb{R}^d if there is *some* function $\phi : \mathbb{R}^d \rightarrow \mathcal{F}$ into *some* space \mathcal{F} with scalar product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ such that

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{F}} \quad \text{for all } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d.$$

In other words

- ▶ k is a kernel if it can be interpreted as a scalar product on some other space.
- ▶ If we substitute $k(\mathbf{x}, \mathbf{x}')$ for $\langle \mathbf{x}, \mathbf{x}' \rangle$ in all SVM equations, we implicitly train a *linear* SVM on the space \mathcal{F} .
- ▶ The SVM still works: It still uses scalar products, just on another space.

The mapping ϕ

- ▶ ϕ has to transform the data into data on which a linear SVM works well.
- ▶ This is usually achieved by choosing \mathcal{F} as a higher-dimensional space than \mathbb{R}^d .

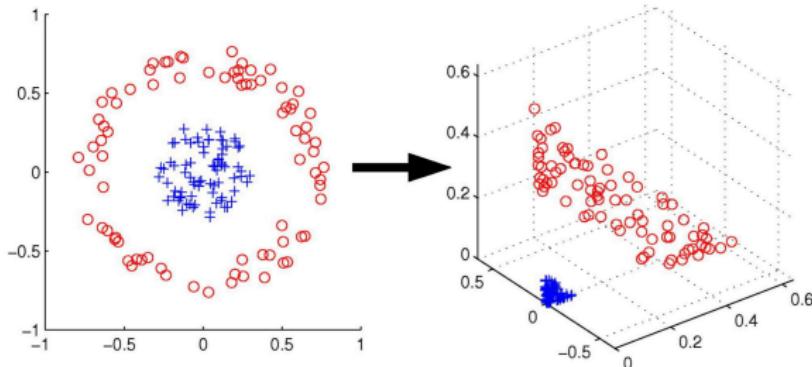
Mapping into Higher Dimensions

Example

How can a map into higher dimensions make class boundary (more) linear?

Consider

$$\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \quad \text{where} \quad \phi \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} x_1^2 \\ 2x_1x_2 \\ x_2^2 \end{pmatrix}$$



Mapping into Higher Dimensions

Problem

In previous example: We have to know what the data looks like to choose ϕ !

Solution

- ▶ Choose high dimension h for \mathcal{F} .
- ▶ Choose components ϕ_i of $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_h(\mathbf{x}))$ as different nonlinear mappings.
- ▶ If two points differ in \mathbb{R}^d , some of the nonlinear mappings will amplify differences.

The RBF kernel is an extreme case

- ▶ The function k_{RBF} can be shown to be a kernel, however:
- ▶ \mathcal{F} is infinite-dimensional for this kernel.

Determining whether k is a kernel

Mercer's theorem

A mathematical result called *Mercer's theorem* states that, if the function k is positive, i.e.

$$\int_{\mathbb{R}^d \times \mathbb{R}^d} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \geq 0$$

for all functions f , then it can be written as

$$k(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x}) \phi_j(\mathbf{x}')$$

The ϕ_j are functions $\mathbb{R}^d \rightarrow \mathbb{R}$ and $\lambda_i \geq 0$. This means the (possibly infinite) vector $\phi(\mathbf{x}) = (\sqrt{\lambda_1} \phi_1(\mathbf{x}), \sqrt{\lambda_2} \phi_2(\mathbf{x}), \dots)$ is a feature map.

Kernel arithmetic

Various functions of kernels are again kernels: If k_1 and k_2 are kernels, then e.g.

$$k_1 + k_2$$

$$k_1 \cdot k_2$$

$$\text{const.} \cdot k_1$$

are again kernels.

The Kernel Trick

Kernels in general

- ▶ Many linear machine learning and statistics algorithms can be "kernelized".
- ▶ The only conditions are:
 1. The algorithm uses a scalar product.
 2. In all relevant equations, the data (and all other elements of \mathbb{R}^d) appear *only inside a scalar product*.
- ▶ This approach to making algorithms non-linear is known as the "kernel trick".

Kernel SVM

Optimization problem

$$\begin{aligned} \min_{\mathbf{v}_H, c} \quad & \|\mathbf{v}_H\|_{\mathcal{F}}^2 + \gamma \sum_{i=1}^n \xi_i^2 \\ \text{s.t.} \quad & \tilde{y}_i (\langle \mathbf{v}_H, \phi(\tilde{\mathbf{x}}_i) \rangle_{\mathcal{F}} - c) \geq 1 - \xi_i \quad \text{and } \xi_i \geq 0 \end{aligned}$$

Note: \mathbf{v}_H now lives in \mathcal{F} , and $\|\cdot\|_{\mathcal{F}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ are norm and scalar product on \mathcal{F} .

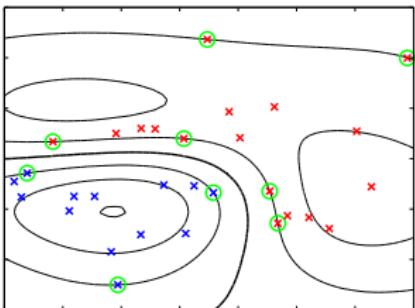
Dual optimization problem

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & W(\alpha) := \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j (\mathbf{k}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)) + \frac{1}{\gamma} \mathbb{I}\{i = j\} \\ \text{s.t.} \quad & \sum_{i=1}^n \tilde{y}_i \alpha_i = 0 \quad \text{and} \quad \alpha_i \geq 0 \end{aligned}$$

Classifier

$$f(\mathbf{x}) = \operatorname{sgn} \left(\sum_{i=1}^n \tilde{y}_i \alpha_i^* \mathbf{k}(\tilde{\mathbf{x}}_i, \mathbf{x}) - c \right)$$

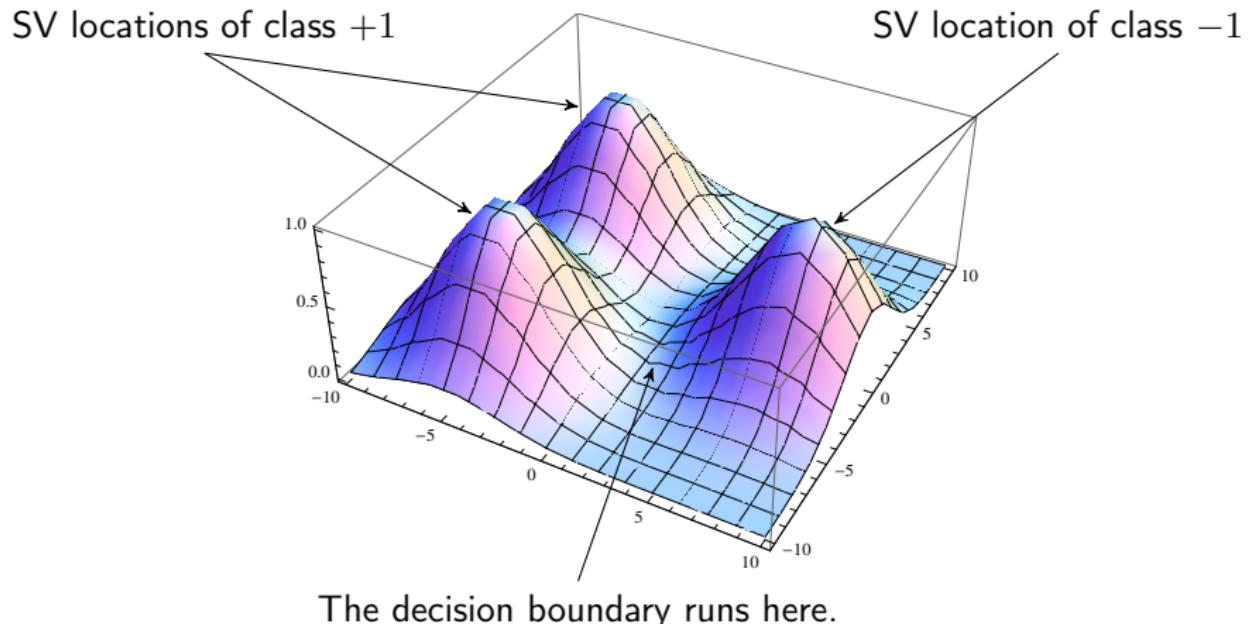
SVM with RBF Kernel



$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^n y_i \alpha_i^* k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}) \right)$$

- ▶ Circled points are support vectors. The two contour lines running through support vectors are the nonlinear counterparts of the convex hulls.
- ▶ The thick black line is the classifier.
- ▶ Think of a Gaussian-shaped function $k_{\text{RBF}}(\cdot, \mathbf{x}')$ centered at each support vector \mathbf{x}' . These functions add up to a function surface over \mathbb{R}^2 .
- ▶ The lines in the image are contour lines of this surface. The classifier runs along the bottom of the "valley" between the two classes.
- ▶ Smoothness of the contours is controlled by σ

Decision Boundary with RBF Kernel



The decision boundary of the classifier coincides with the set of points where the surfaces for class +1 and class -1 have equal value.

Summary: SVMs

Basic SVM

- ▶ Linear classifier for linearly separable data.
- ▶ Positions of affine hyperplane is determined by maximizing margin.
- ▶ Maximizing the margin is a convex optimization problem.

Full-fledged SVM

Ingredient	Purpose
Maximum margin	Good generalization properties
Slack variables	Overlapping classes
Kernel	Robustness against outliers Nonlinear decision boundary

Use in practice

- ▶ Software packages (e.g. libsvm, SVMLite)
- ▶ Choose a kernel function (e.g. RBF)
- ▶ Cross-validate margin parameter γ and kernel parameters (e.g. bandwidth)

Lecture 10: Cross validation

Reading: Sections 7.10, 7.11

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
February 23, 2018

Validation set approach

Goal:

- ▶ Estimate the test error of a learning method.
- ▶ Select the best model from a given set of models. “Set of models” can simply mean “set of different parameter values”.

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 2

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 2

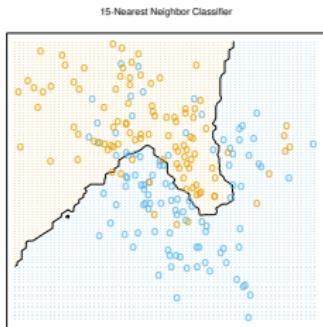


FIGURE 2.2. The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (**BLUE** = 0, **ORANGE** = 1) and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.

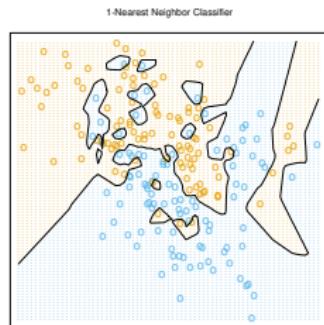


FIGURE 2.3. The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (**BLUE** = 0, **ORANGE** = 1), and then predicted by 1-nearest-neighbor classification.

Training Vs. test error

Training error IS NOT a good estimate of the test error.

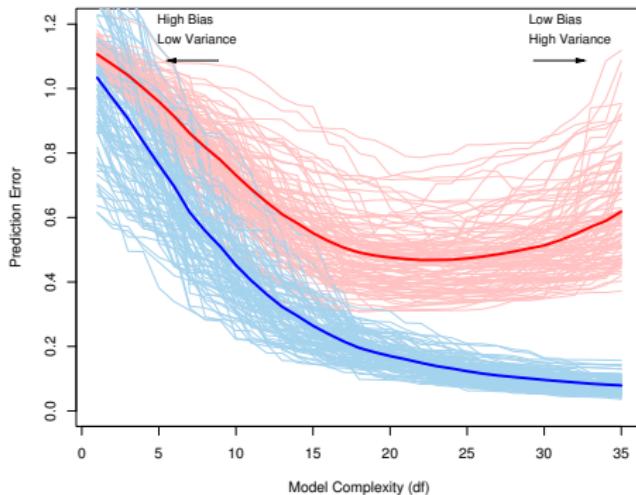


FIGURE 7.1. Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error \bar{err} , while the light red curves show the conditional test error Err_T for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error Err and the expected training error $E[\bar{err}]$.

Some concepts

$L(\cdot, \cdot)$ is the loss function.

Training error is the average loss over the training sample

$$\overline{\text{err}} = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i)).$$

Test error, also referred to as **generalization error**, is the prediction error over an independent test sample

$$\text{Err}_{\mathcal{T}} = \mathbb{E}[L(Y, \hat{f}(X)) | \mathcal{T}].$$

Usually, it is more amenable to estimate the **expected prediction error** (or expected test error)

$$\text{Err} = \mathbb{E}[L(Y, \hat{f}(X))] = \mathbb{E}[\text{Err}_{\mathcal{T}}].$$

Model selection

Model selection: estimating the performance of different models in order to choose the best one.

- ▶ Randomly split data into three sets: a training set, a validation set and a test set.



- ▶ Train different models on the training set.
- ▶ Evaluate each trained model on the validation set (i.e. compute prediction error).
- ▶ Select the model with lowest prediction error.

Model assessment: having chosen a final model, estimating its prediction error (generalization error) on new data.

- ▶ Finally, estimate the prediction error of the selected model on the test set.

K-fold cross-validation

Each of the error estimates computed on validation set is computed from a single example of a trained classifier. Can we improve the estimate?

Strategy:

- ▶ Set aside the test set.
- ▶ Split the remaining data into K blocks.
- ▶ Use each block in turn as validation set. Perform cross validation and average the results over all K combinations.

This method is called *K*-fold cross-validation.

Example: $K=5$, step $k=3$

1	2	3	4	5
Train	Train	Validation	Train	Train

K-fold cross-validation

Assume we have a set of models $f(\cdot, \alpha)$ indexed by a tuning parameter α .

Estimating prediction error

- ▶ Split data into K equally sized blocks.
- ▶ Train an instance $\hat{f}^{-k}(\cdot, \alpha)$ of the model, using all blocks except block k as training data.
- ▶ Compute the cross validation estimate

$$CV(\hat{f}, \alpha) := \frac{1}{K} \sum_{k=1}^K \frac{1}{|\text{block } k|} \sum_{(x,y) \in \text{block } k} L(y, \hat{f}^{-k}(x, \alpha))$$

Repeat this for each tuning parameter α .

Selecting a model

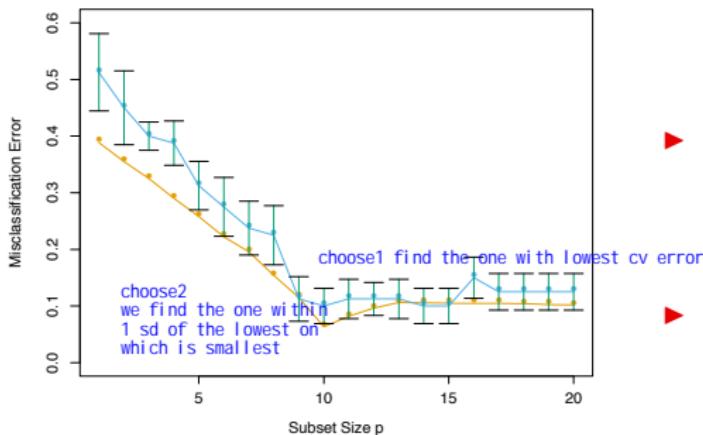
Choose the parameter value α^* according to certain criteria.

Model assessment

Report risk estimate for $f(\cdot, \alpha^*)$ computed on test data.

The one standard error rule

Forward stepwise selection



Blue: 10-fold cross validation

Yellow: True test error

- ▶ A number of models with $10 \leq p \leq 15$ have the same CV error.
- ▶ The vertical bars represent 1 standard error in the test error from the 10 folds.
- ▶ **Rule of thumb:** Choose the simplest model whose CV error is no more than one standard error above the model with the lowest CV error.

How to choose K

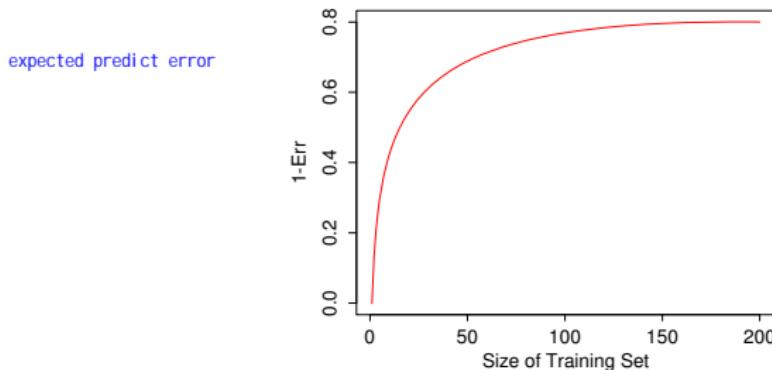
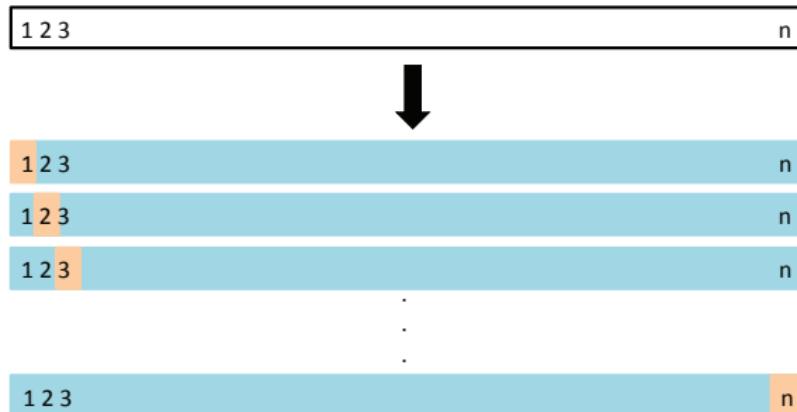


FIGURE 7.8. Hypothetical learning curve for a classifier on a given task: a plot of $1 - \text{Err}$ versus the size of the training set N . With a dataset of 200 observations, 5-fold cross-validation would use training sets of size 160, which would behave much like the full set. However, with a dataset of 50 observations fivefold cross-validation would use training sets of size 40, and this would result in a considerable overestimate of prediction error.

Leave one out cross-validation

- ▶ For every $i = 1, \dots, n$:
 - ▶ train the model on every point except i ,
 - ▶ compute the test error on the held out point.
- ▶ Average the test errors.



Leave one out cross-validation

- ▶ For every $i = 1, \dots, n$:
 - ▶ train the model on every point except i ,
 - ▶ compute the test error on the held out point.
- ▶ Average the test errors.

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i^{(-i)})^2$$

Prediction for the i sample without using the i th sample.

Leave one out cross-validation

- ▶ For every $i = 1, \dots, n$:
 - ▶ train the model on every point except i ,
 - ▶ compute the test error on the held out point.
- ▶ Average the test errors.

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(y_i \neq \hat{y}_i^{(-i)})$$

... for a classification problem.

Leave one out cross-validation

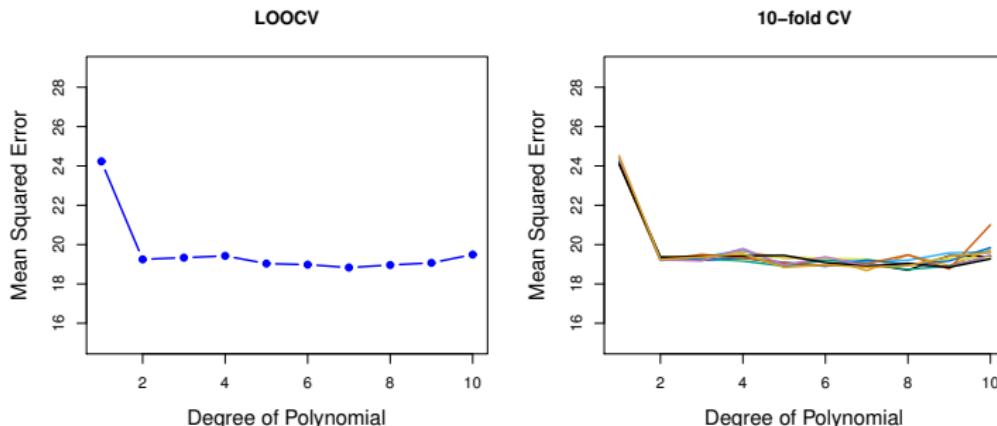
Computing $\text{CV}_{(n)}$ can be computationally expensive, since it involves fitting the model n times.

For linear regression, there is a shortcut:

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2$$

where h_{ii} is the leverage statistic.

LOOCV vs. K -fold cross-validation



- ▶ K -fold CV depends on the chosen split.
- ▶ In K -fold CV, we train the model on less data than what is available. This introduces **bias** into the estimates of test error.
- ▶ In LOOCV, the training samples highly resemble each other. This increases the **variance** of the test error estimate.

The wrong way to do cross validation

Reading: Section 7.10.2 of The Elements of Statistical Learning.

We want to classify 200 individuals according to whether they have cancer or not. We use logistic regression onto 1000 measurements of gene expression.

这个问题有两个步骤都涉及到了数据，第一个步骤是变量的选取（相当于要降维），选取的结果是基于所用的数据的。第二个是使用logistics回归，对参数进行估计。这个估计的结果也和训练的数据有关的。

这里只有第二部用到了cross validation，第一步直接用的是全部的数据。但是问题在于，全部的数据下，样本量远小于维度。其实很难保证各个维度之间的独立性。

Proposed strategy:

- ▶ Using all the data, select the 20 most significant genes using z -tests.
- ▶ Estimate the test error of logistic regression with these 20 predictors via 10-fold cross validation.

The wrong way to do cross validation

To see how that works, let's use the following simulated data:

- ▶ Each gene expression is standard normal and independent of all others.
- ▶ The response (cancer or not) is sampled from a coin flip — no correlation to any of the “genes”.

What should the misclassification rate be for any classification method using these predictors?

Roughly 50%.

The wrong way to do cross validation

We run this simulation, and obtain a CV error rate of 3%!

Why is this?

- ▶ Since we only have 200 individuals in total, among 1000 variables, at least some will be correlated with the response.
数据不足以体现各个attribute之间的区别，很可能各个attribute之间会有很大的相关性
- ▶ We do variable selection using *all the data*, so the variables we select have some correlation with the response in every subset or fold in the cross validation.

因为我们用的是所有的数据进行的降维。所以认为的引入了一些潜在的信息，这些信息提高了数据之间的相关性，从而降低了training error。但是不能保证test error

The **right** way to do cross validation

- ▶ Divide the data into 10 folds.
- ▶ For $i = 1, \dots, 10$:
 - ▶ Using every fold except i , perform the variable selection and fit the model with the selected variables.
 - ▶ Compute the error on fold i .
- ▶ Average the 10 test errors obtained.

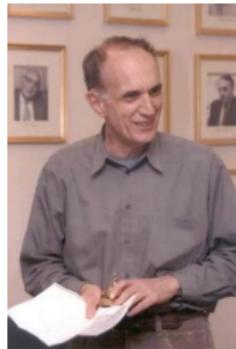
In our simulation, this produces an error estimate of close to 50%.

Moral of the story: Every aspect of the learning method that involves using the data — variable selection, for example — must be cross-validated.

Cross-validation vs. the Bootstrap

Cross-validation: provides estimates of the (test) error.

The Bootstrap: provides the (standard) error of estimates.
variance



- ▶ One of the most important techniques in all of Statistics.
- ▶ Computer intensive method.
- ▶ Popularized by Brad Efron.

Standard errors in linear regression

Standard error: SD of an estimate from a sample of size n .

```
Residuals:
    Min      1Q  Median      3Q     Max 
-15.594 -2.730 -0.518  1.777 26.199 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3.646e+01 5.103e+00 7.144 3.28e-12 ***
crim        -1.080e-01 3.286e-02 -3.287 0.001087 ** 
zn          4.642e-02 1.373e-02 3.382 0.000778 *** 
indus       2.056e-02 6.150e-02 0.334 0.738288    
chas        2.687e+00 8.616e-01 3.118 0.001925 ** 
nox         -1.777e+01 3.820e+00 -4.651 4.25e-06 ***
rm          3.810e+00 4.179e-01 9.116 < 2e-16 ***
age         6.922e-04 1.321e-02 0.052 0.958229    
dis         -1.476e+00 1.995e-01 -7.398 6.01e-13 ***
rad         3.060e-01 6.635e-02 4.613 5.07e-06 ***
tax         -1.233e-02 3.761e-03 -3.280 0.001112 ** 
ptratio     -9.527e-01 1.308e-01 -7.283 1.31e-12 *** 
black       9.312e-03 2.686e-03 3.467 0.000573 *** 
lstat      -5.248e-01 5.072e-02 -10.347 < 2e-16 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-Squared: 0.7406,      Adjusted R-squared: 0.7338 
F-statistic: 108.1 on 13 and 492 DF,   p-value: < 2.2e-16
```

Classical way to compute Standard Errors

Example: Estimate the variance of a sample x_1, x_2, \dots, x_n :

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2.$$

What is the **Standard Error of $\hat{\sigma}^2$** ?

- ▶ Assume that x_1, \dots, x_n are normally distributed.
- ▶ Assume that the true variance is close to $\hat{\sigma}^2$ and the true mean is close to \bar{x} .
- ▶ Then $\hat{\sigma}^2(n-1)$ has a χ^2 -squared distribution with n degrees of freedom.
- ▶ The SD of this *sampling distribution* is the Standard Error.

Limitations of the classical approach

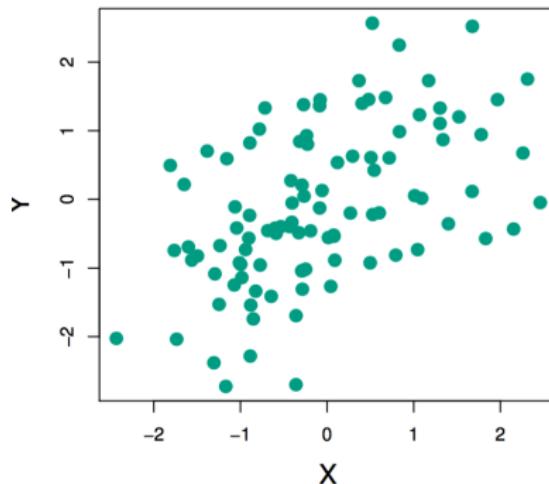
This approach has served statisticians well for 90 years; however, what happens if:

- ▶ The distributional assumption — for example, x_1, \dots, x_n being normal — breaks down?
- ▶ The estimator does not have a simple form and its sampling distribution cannot be derived analytically?

Example. Investing in two assets

Suppose that X and Y are the returns of two assets.

These returns are observed every day: $(x_1, y_1), \dots, (x_n, y_n)$.



Example. Investing in two assets

We have a fixed amount of money to invest and we will invest a fraction α on X and a fraction $(1 - \alpha)$ on Y . Therefore, our return will be

$$\alpha X + (1 - \alpha)Y.$$

Our goal will be to minimize the variance of our return as a function of α . One can show that the optimal α is:

$$\alpha = \frac{\sigma_Y^2 - \text{Cov}(X, Y)}{\sigma_X^2 + \sigma_Y^2 - 2\text{Cov}(X, Y)}.$$

如果我们有很多组独立的数据，那么每一组的数据，我们都可以获得一个alpha的估计值。这样就可以得到一组alpha。利用这一组的alpha，我们就可以近似出alpha的分布，计算出alpha_hat的分布

Proposal: Use an estimate:

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\text{Cov}}(X, Y)}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\text{Cov}}(X, Y)}.$$

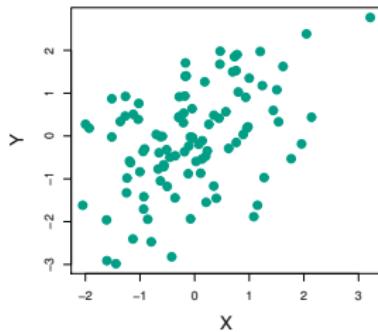
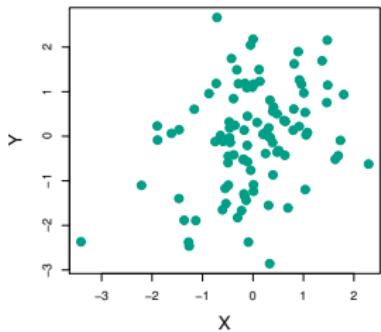
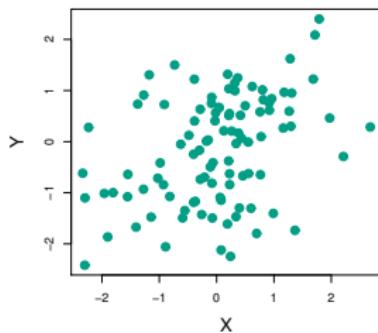
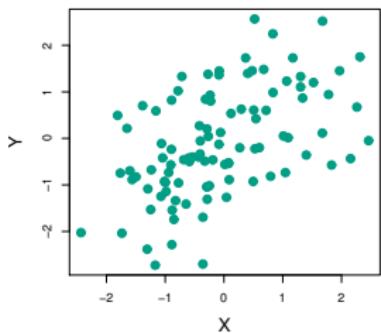
Example. Investing in two assets

Suppose we compute the estimate $\hat{\alpha} = 0.6$ using the samples $(x_1, y_1), \dots, (x_n, y_n)$.

- ▶ How sure can we be of this value?
- ▶ If we resampled the observations, would we get a wildly different $\hat{\alpha}$?

In this thought experiment, we know the actual joint distribution $P(X, Y)$, so we can resample the n observations to our hearts' content.

Resampling the data from the true distribution



Computing the standard error of $\hat{\alpha}$

For each resampling of the data,

$$(x_1^{(1)}, \dots, x_n^{(1)})$$

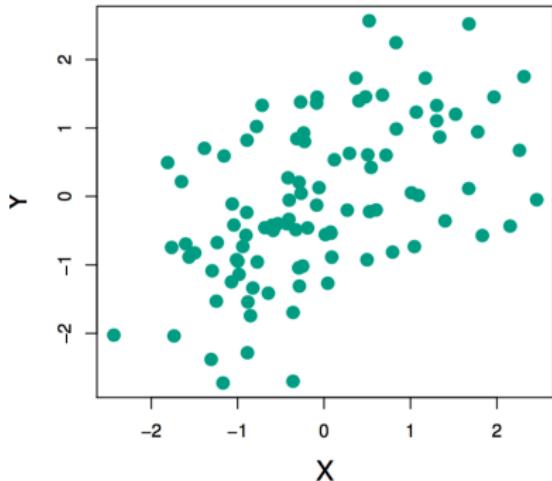
$$(x_1^{(2)}, \dots, x_n^{(2)})$$

...

we can compute a value of the estimate $\hat{\alpha}^{(1)}, \hat{\alpha}^{(2)}, \dots$

The Standard Error of $\hat{\alpha}$ is approximated by the standard deviation of these values.

In reality, we only have n samples

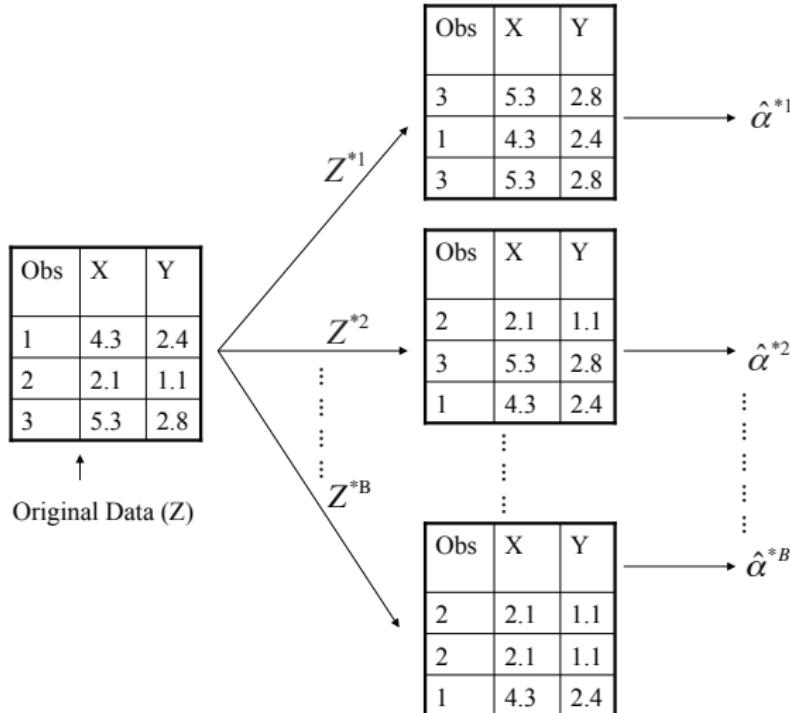


- ▶ However, these samples can be used to approximate the joint distribution of X and Y .
- ▶ **The Bootstrap:** Resample from the *empirical distribution*:

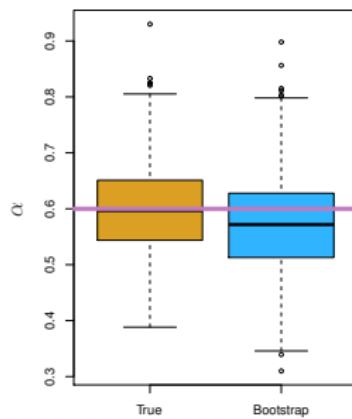
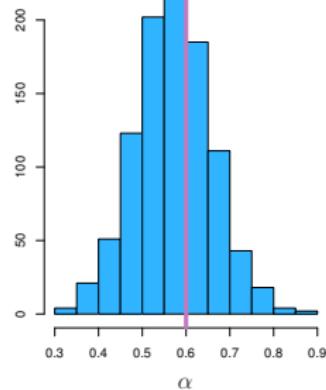
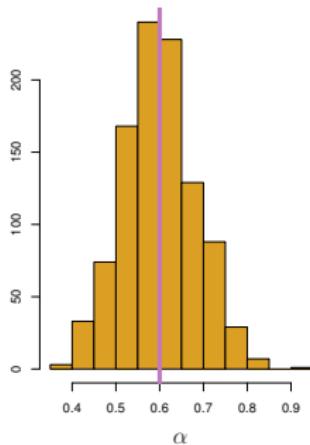
$$\hat{P}(X, Y) = \frac{1}{n} \sum_{i=1}^n \delta(x_i, y_i).$$

- ▶ Equivalently, resample the data by drawing n samples *with replacement* from the actual observations.

A schematic of the Bootstrap



Comparing Bootstrap resamplings to resamplings from the true distribution



Lecture 11: Model selection and regularization

Reading: Sections 7.4 - 7.7, 3.4

GU4241/GR5241 Statistical Machine Learning

Linxí Liu
February 23, 2018

What do we know so far

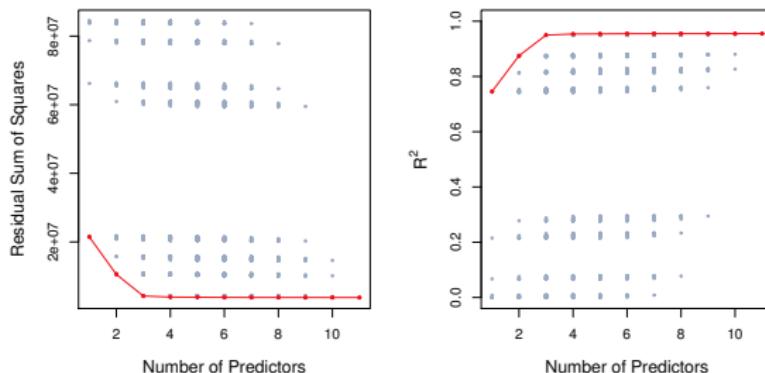
- ▶ In linear regression, adding predictors always decreases the training error or RSS.
- ▶ However, adding predictors does not necessarily improve the test error.
- ▶ Selecting significant predictors is hard when n is not much larger than p .
- ▶ When $n < p$, there is no least squares solution:

$$\hat{\beta} = \underbrace{(\mathbf{X}^T \mathbf{X})^{-1}}_{\text{Singular}} \mathbf{X}^T \mathbf{y}.$$

So, we must find a way to select fewer predictors.

Best subset selection

- ▶ Simple idea: let's compare all models with k predictors.
- ▶ There are $\binom{p}{k} = p! / [k!(p - k)!]$ possible models.
- ▶ Choose the model with the smallest RSS. Do this for every possible k .



- ▶ Naturally, the RSS and R^2 improve as we increase k .

Best subset selection

To optimize k , we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. Akaike Information Criterion (AIC):

$$\frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2k\hat{\sigma}^2)$$

后面的部分是penalty

where $\hat{\sigma}^2$ is an estimate of the irreducible error.

Best subset selection

To optimize k , we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. Akaike Information Criterion (AIC) or C_p :

$$\frac{1}{n}(\text{RSS} + 2k\hat{\sigma}^2)$$

where $\hat{\sigma}^2$ is an estimate of the irreducible error.

for different model there will have different form of AIC and BIC

Best subset selection

To optimize k , we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. Akaike Information Criterion (AIC) or C_p :
2. Bayesian Information Criterion (BIC):

$$\frac{1}{n}(\text{RSS} + \log(n)k\hat{\sigma}^2)$$

BIC得到的值比较小， simpler model

Best subset selection

To optimize k , we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. Akaike Information Criterion (AIC) or C_p :
2. Bayesian Information Criterion (BIC):
3. Adjusted R^2 :

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

Best subset selection

To optimize k , we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

1. Akaike Information Criterion (AIC) or C_p :
2. Bayesian Information Criterion (BIC):
3. Adjusted R^2 :

$$R_{\text{adj}}^2 = 1 - \frac{\text{RSS}/(n - k - 1)}{\text{TSS}/(n - 1)}$$

Best subset selection

To optimize k , we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

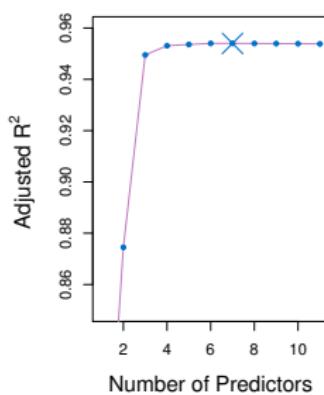
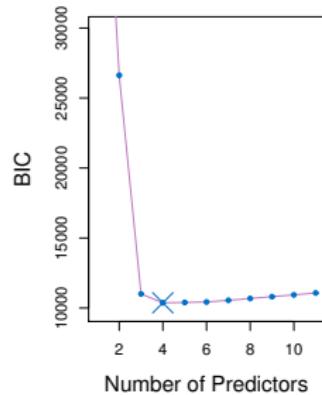
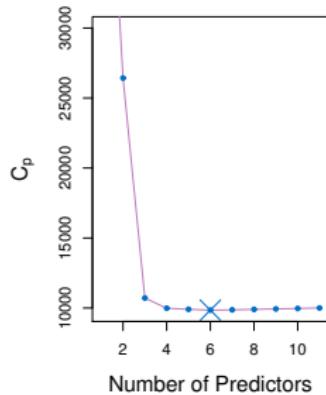
1. Akaike Information Criterion (AIC) or C_p :
2. Bayesian Information Criterion (BIC):
3. Adjusted R^2 :

How do they compare to cross validation:

- ▶ They are **much less expensive to compute**.
- ▶ They are motivated by **asymptotic arguments and rely on model assumptions** (e.g. normality of the errors).
- ▶ Equivalent concepts for other models (e.g. logistic regression).

Example

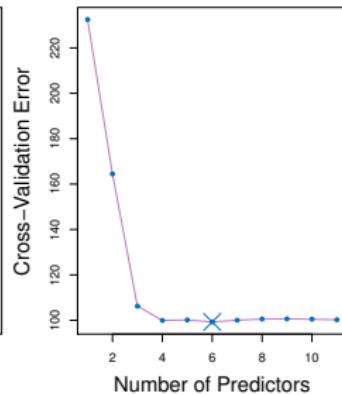
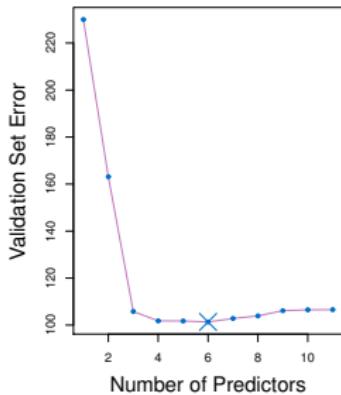
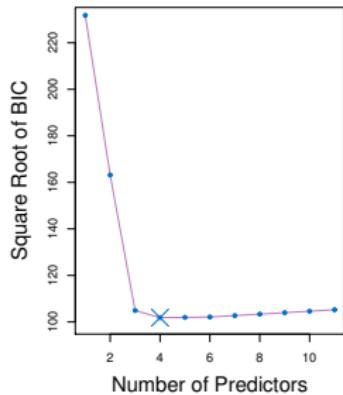
Best subset selection for the Credit dataset.



In linear adjusted R square is just equal to cp

Example

Cross-validation vs. the BIC.



Recall: In k -fold cross validation, we can estimate a standard error or accuracy for our test error estimate. Then, we apply the one standard-error rule.

Stepwise selection methods

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k , there are too many possibilities, we increase our chances of overfitting. **The model selected has *high variance*.**

In order to mitigate these problems, we can restrict our search space for the best model.

This reduces the variance of the selected model at the expense of an increase in bias.

Forward selection

Algorithm 6.2 *Forward stepwise selection*

1. Let \mathcal{M}_0 denote the *null* model, which contains no predictors.
 2. For $k = 0, \dots, p - 1$:
 - (a) Consider all $p - k$ models that augment the predictors in \mathcal{M}_k with one additional predictor.
 - (b) Choose the *best* among these $p - k$ models, and call it \mathcal{M}_{k+1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Forward selection vs. best subset

# Variables	Best subset	Forward stepwise
One	rating	rating
Two	rating, income	rating, income
Three	rating, income, student	rating, income, student
Four	cards, income student, limit	rating, income, student, limit

TABLE 6.1. The first four selected models for best subset selection and forward stepwise selection on the Credit data set. The first three models are identical but the fourth models differ.

Backward selection

Algorithm 6.3 Backward stepwise selection

1. Let \mathcal{M}_p denote the *full* model, which contains all p predictors.
 2. For $k = p, p-1, \dots, 1$:
 - (a) Consider all k models that contain all but one of the predictors in \mathcal{M}_k , for a total of $k-1$ predictors.
 - (b) Choose the *best* among these k models, and call it \mathcal{M}_{k-1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Forward vs. backward selection

- ▶ You cannot apply backward selection when $p > n$.
- ▶ Although it seems like they should, they need not produce the same sequence of models.

Example. $X_1, X_2 \sim \mathcal{N}(0, \sigma)$ independent.

$$X_3 = X_1 + 3X_2$$

$$Y = X_1 + 2X_2 + \epsilon$$

Regress Y onto X_1, X_2, X_3 .

- ▶ Forward: $\{X_3\} \rightarrow \{X_3, X_2\} \rightarrow \{X_3, X_2, X_1\}$
- ▶ Backward: $\{X_1, X_2, X_3\} \rightarrow \{X_1, X_2\} \rightarrow \{X_2\}$

Other stepwise selection strategies

- ▶ **Mixed stepwise selection:** Do forward selection, but at every step, remove any variables that are no longer “necessary”.
- ▶ **Forward stagewise selection:** Do forward selection, but after every step, modify the remaining predictors such that they are uncorrelated to the selected predictors.
- ▶ ...

Lecture 12: Shrinkage

Reading: Section 3.4

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
March 2, 2018

Issues with Least Squares

Robustness

- ▶ Least squares works only if \mathbf{X} has full column rank, i.e. if $\mathbf{X}^T \mathbf{X}$ is invertible.
- ▶ If $\mathbf{X}^T \mathbf{X}$ almost not invertible, least squares is numerically unstable.
Statistical consequence: High variance of predictions.

Not suited for high-dimensional data

- ▶ Modern problems: Many dimensions/features/predictors (possibly thousands)
- ▶ Only a few of these may be important
→ need some form of feature selection
- ▶ Least squares:
 - ▶ Treats all dimensions equally
 - ▶ Relevant dimensions are averaged with irrelevant ones
 - ▶ Consequence: Signal loss

Regularity of Matrices

Regularity

A matrix which is not invertible is also called a **singular** matrix. A matrix which is invertible (not singular) is called **regular**.

In computations

Numerically, matrices can be "almost singular". Intuition:

- ▶ A singular matrix maps an entire linear subspace into a single point.
- ▶ If a matrix maps points far away from each other to points very close to each other, it almost behaves like a singular matrix.

Regularity of Symmetric Matrices

A positive semi-definite matrix A is singular \Leftrightarrow smallest EValue is 0

Illustration

If smallest EValue $\lambda_{\min} > 0$ but very small (say $\lambda_{\min} \approx 10^{-10}$):

- ▶ Suppose x_1, x_2 are two points in subspace spanned by ξ_{\min} with $\|x_1 - x_2\| \approx 1000$.
eigenvector 起的是放缩作用
- ▶ Image under A : $\|Ax_1 - Ax_2\| \approx 10^{-7}$

In this case

- ▶ A has an inverse, but A behaves almost like a singular matrix
- ▶ The inverse A^{-1} can map almost identical points to points with large distance, i.e.

small change in input \rightarrow large change in output

Consequence for Statistics

If a statistical prediction involves the inverse of an almost-singular matrix, the predictions become unreliable (high variance).

Implications for Linear Regression

Recall: Prediction in linear regression

For a point $\mathbf{x}_{\text{new}} \in \mathbb{R}^p$, we predict the corresponding function value as

$$\hat{y}_{\text{new}} = \langle \hat{\beta}, \mathbf{x}_{\text{new}} \rangle = \mathbf{x}_{\text{new}} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Effect of unstable inversion

- ▶ Suppose we choose an arbitrary training point \mathbf{x}_i and make a small change to its response value y_i .
- ▶ Intuitively, that should not have a big impact on $\hat{\beta}$ or on prediction.
- ▶ If $\mathbf{X}^T \mathbf{X}$ is almost singular, a small change to y_i can prompt a huge change in $\hat{\beta}$, and hence in the predicted value \hat{y}_{new} .

Measuring Regularity (of Symmetric Matrices)

Symmetric matrices

Denote by λ_{\max} and λ_{\min} the eigenvalues of A with largest/smallest absolute value. If A is symmetric, then

$$A \text{ regular} \iff |\lambda_{\min}| > 0.$$

Idea

- We can use $|\lambda_{\min}|$ as a measure of regularity:

larger value of λ_{\min} \leftrightarrow "more regular" matrix A

- We need a notion of scale to determine whether $|\lambda_{\min}|$ is large.
- The relevant scale is how A scales a vector. Maximal scaling coefficient: λ_{\max} .

Regularity measure

$$c(A) := \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

The function $c(\cdot)$ is called the **spectral condition** ("spectral" since the set of eigenvalues is also called the "spectrum").

Ridge Regression

Objective

Ridge regression is a modification of least squares. We try to make least squares more robust if $\mathbf{X}^T \mathbf{X}$ is almost singular.

Ridge regression solution

The ridge regression solution to a linear regression problem is defined as

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} := (\mathbf{X}^T \mathbf{X} + \lambda \mathbb{I})^{-1} \mathbf{X}^T \mathbf{y}$$

λ is a tuning parameter.

Explanation

Recall

$\mathbf{X}^T \mathbf{X} \in \mathbb{R}^{p \times p}$ is positive definite.

Spectral shift

Suppose ξ_1, \dots, ξ_p are EVectors of $\mathbf{X}^T \mathbf{X}$ with EValues $\lambda_1, \dots, \lambda_p$.
Then:

$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbb{I})\xi_i = (\mathbf{X}^T \mathbf{X})\xi_i + \lambda \mathbb{I}\xi_i = (\lambda_i + \lambda)\xi_i$$

Hence: $(\mathbf{X}^T \mathbf{X} + \lambda \mathbb{I})$ is positive definite with EValues $\lambda_1 + \lambda, \dots, \lambda_p + \lambda$.

Conclusion

$\mathbf{X}^T \mathbf{X} + \lambda \mathbb{I}$ is a *regularized* version of $\mathbf{X}^T \mathbf{X}$.

Implications for statistics

Effect of regularization

- ▶ We deliberately distort prediction:
 - ▶ If least squares ($\lambda = 0$) predicts perfectly, the ridge regression prediction has an error that increases with λ .
 - ▶ Hence: Biased estimator, bias increases with λ .
- ▶ Spectral shift regularizes matrix → decreases variance of predictions.

Bias-variance trade-off

- ▶ We decrease the variance (improve robustness) at the price of incurring a bias.
- ▶ λ controls the trade-off between bias and variance.

Cost Function

- ▶ Linear regression solution was defined as minimizer of $L(\beta) := \|\mathbf{y} - \mathbf{X}\beta\|^2$
- ▶ We have so far defined ridge regression only directly in terms of the estimator $\hat{\beta}^{\text{ridge}} := (\mathbf{X}^T \mathbf{X} + \lambda \mathbb{I})^{-1} \mathbf{X}^T \mathbf{y}$.
- ▶ To analyze the method, it is helpful to understand it as an optimization problem.
- ▶ We ask: Which function L' does $\hat{\beta}^{\text{ridge}}$ minimize?

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β , or $\|\beta\|_2^2$. It is called a **penalty term**.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β , or $\|\beta\|_2^2$. It is called a **penalty term**.

The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β , or $\|\beta\|_2^2$. It is called a **penalty term**.

The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

We find an estimate $\hat{\beta}_\lambda^{\text{ridge}}$ for many values of λ and then choose it by cross-validation.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β , or $\|\beta\|_2^2$. It is called a **penalty term**.

The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

We find an estimate $\hat{\beta}_\lambda^{\text{ridge}}$ for many values of λ and then choose it by cross-validation. **Fortunately, this is no more expensive than running a least-squares regression.**

Ridge regression

In least-squares linear regression, scaling the variables has no effect on the fit of the model:

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c ,
ie. after doing this we have the same RSS.

Ridge regression

In least-squares linear regression, scaling the variables has no effect on the fit of the model:

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c , ie. after doing this we have the same RSS.

In ridge regression, this is not true.

Ridge regression

In least-squares linear regression, scaling the variables has no effect on the fit of the model:

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c , ie. after doing this we have the same RSS.

In ridge regression, this is not true.

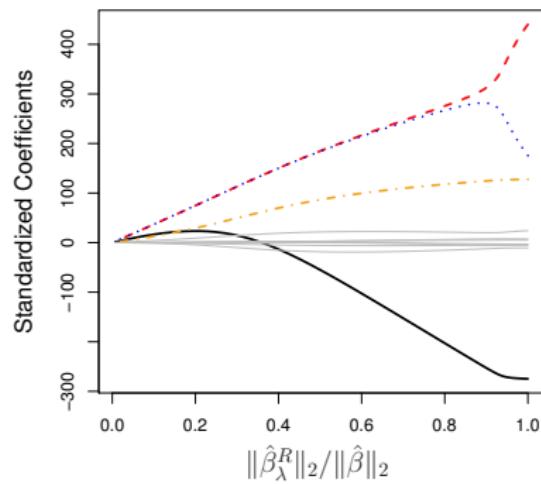
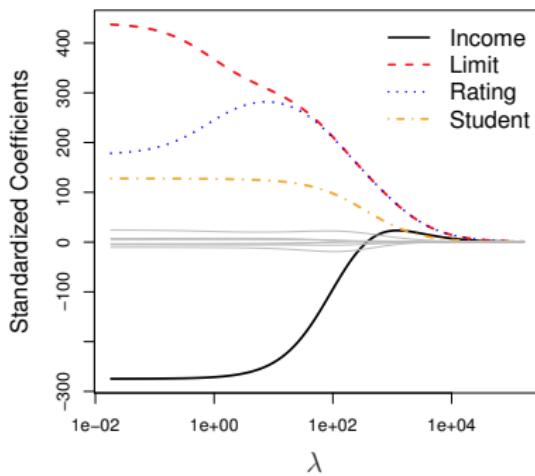
In practice, what do we do?

岭回归需要标准化

- ▶ Scale each variable such that it has sample variance 1 before running the regression.
- ▶ This prevents penalizing some coefficients more than others.

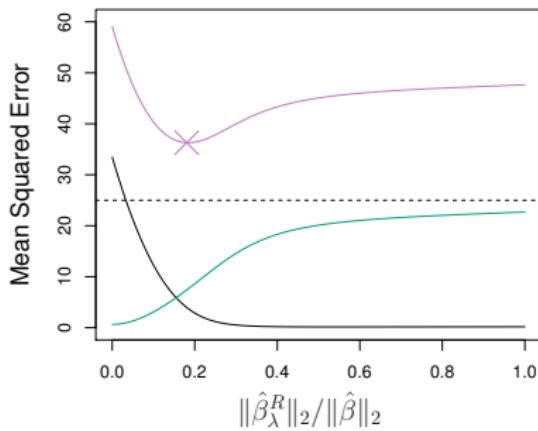
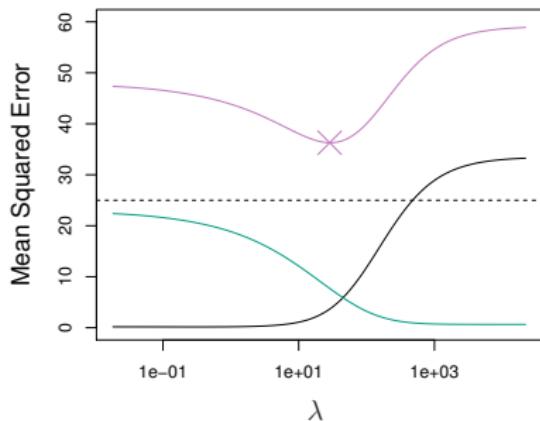
Example. Ridge regression

Ridge regression of default in the Credit dataset.



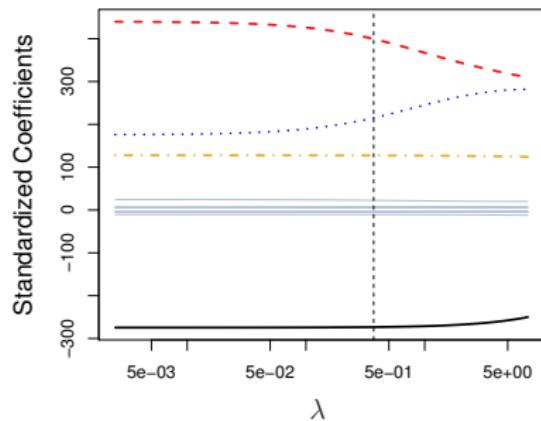
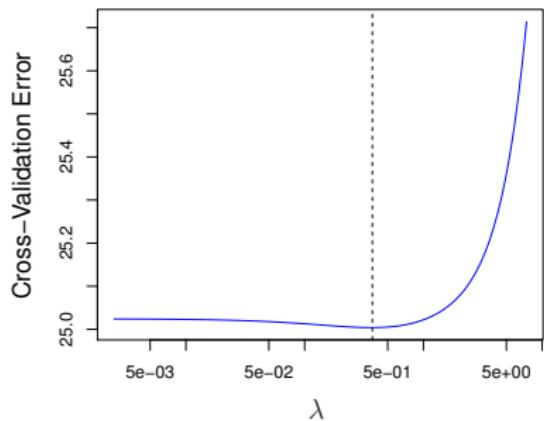
Bias-variance tradeoff

In a simulation study, we compute bias, variance, and test error as a function of λ .



Cross validation would yield an estimate of the test error.

Selecting λ by cross-validation



The Lasso

Lasso regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

In blue, we have the RSS of the model.

In red, we have the ℓ_1 norm of β , or $\|\beta\|_1$.

The Lasso

Lasso regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

In blue, we have the RSS of the model.

In red, we have the ℓ_1 norm of β , or $\|\beta\|_1$.

Why would we use the Lasso instead of Ridge regression?

The Lasso

Lasso regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

In blue, we have the RSS of the model.

In red, we have the ℓ_1 norm of β , or $\|\beta\|_1$.

Why would we use the Lasso instead of Ridge regression?

- ▶ Ridge regression shrinks all the coefficients to a non-zero value.
LASSO will force some of the coefficients to be zero
but in ridge regression even these coefficients are small
they are not zero.

The Lasso

Lasso regression solves the following optimization:

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

In blue, we have the RSS of the model.

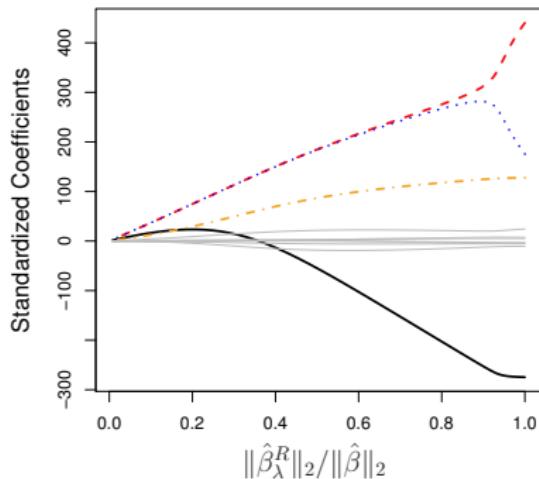
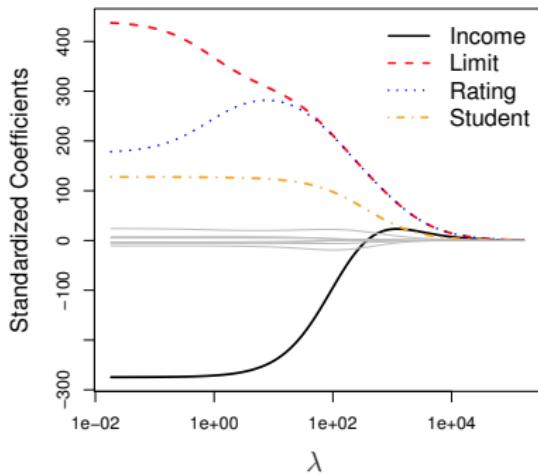
In red, we have the ℓ_1 norm of β , or $\|\beta\|_1$.

Why would we use the Lasso instead of Ridge regression?

- ▶ Ridge regression shrinks all the coefficients to a non-zero value.
- ▶ The Lasso shrinks some of the coefficients all the way to zero.
Alternative to best subset selection or stepwise selection!

Example. Ridge regression

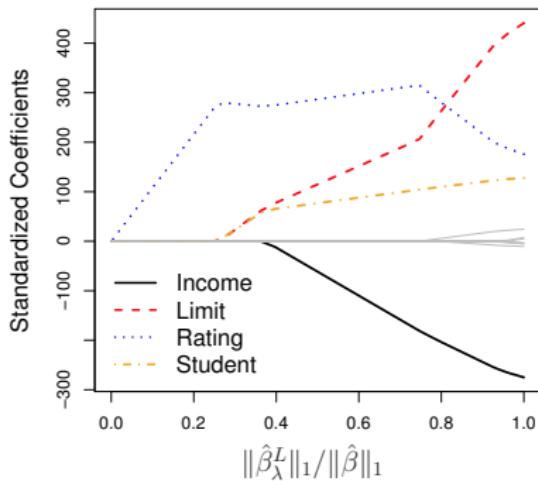
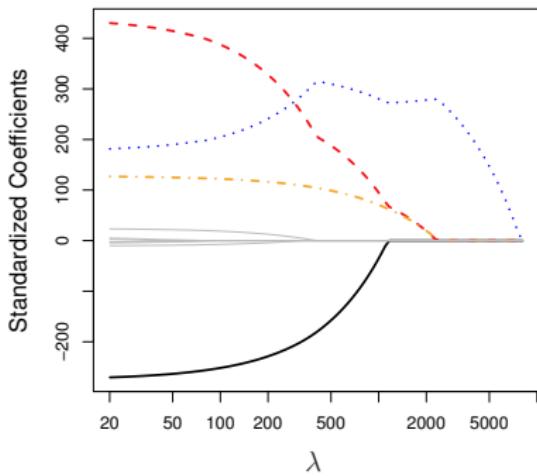
Ridge regression of default in the Credit dataset.



A lot of pesky small coefficients throughout the regularization path.

Example. The Lasso

Lasso regression of default in the Credit dataset.



Those coefficients are shrunk to zero.

An alternative formulation for regularization

- **Ridge:** for every λ , there is an s such that $\hat{\beta}_\lambda^{\text{ridge}}$ solves:

$$\underset{\beta}{\text{minimize}} \quad \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 < s.$$

An alternative formulation for regularization

- **Ridge:** for every λ , there is an s such that $\hat{\beta}_\lambda^{\text{ridge}}$ solves:

$$\underset{\beta}{\text{minimize}} \quad \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 < s.$$

- **Lasso:** for every λ , there is an s such that $\hat{\beta}_\lambda^{\text{lasso}}$ solves:

$$\underset{\beta}{\text{minimize}} \quad \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p |\beta_j| < s.$$

An alternative formulation for regularization

- **Ridge:** for every λ , there is an s such that $\hat{\beta}_\lambda^{\text{ridge}}$ solves:

$$\underset{\beta}{\text{minimize}} \quad \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 < s.$$

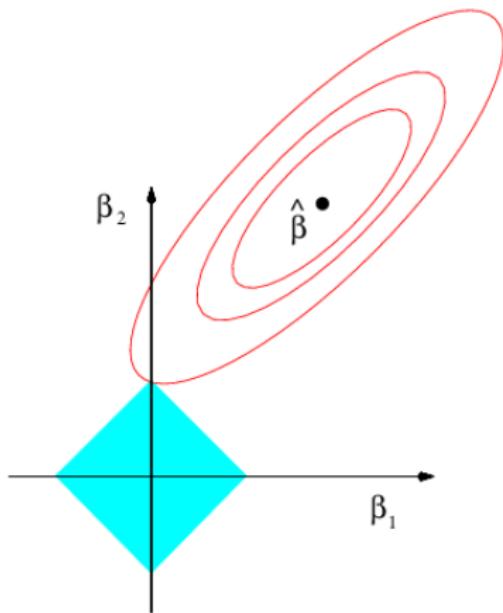
- **Lasso:** for every λ , there is an s such that $\hat{\beta}_\lambda^{\text{lasso}}$ solves:

$$\underset{\beta}{\text{minimize}} \quad \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p |\beta_j| < s.$$

- **Best subset:**

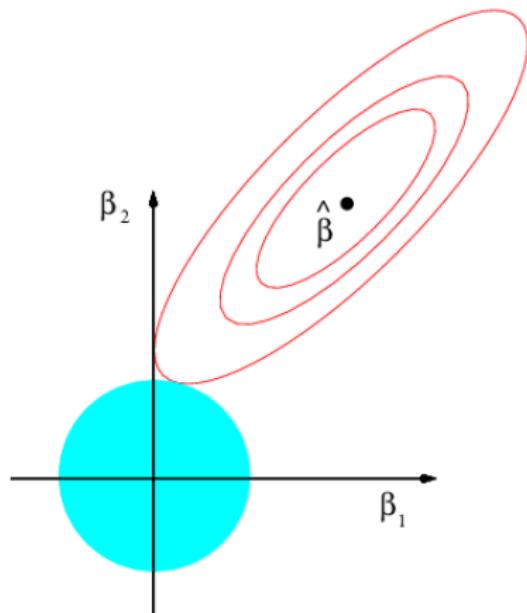
$$\underset{\beta}{\text{minimize}} \quad \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right\} \quad \text{s.t.} \quad \sum_{j=1}^p \mathbf{1}(\beta_j \neq 0) < s.$$

Visualizing Ridge and the Lasso with 2 predictors



The Lasso

$$\blacklozenge : \sum_{j=1}^p |\beta_j| \leq s$$

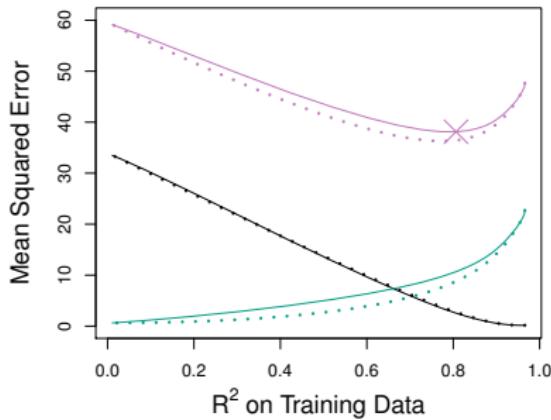
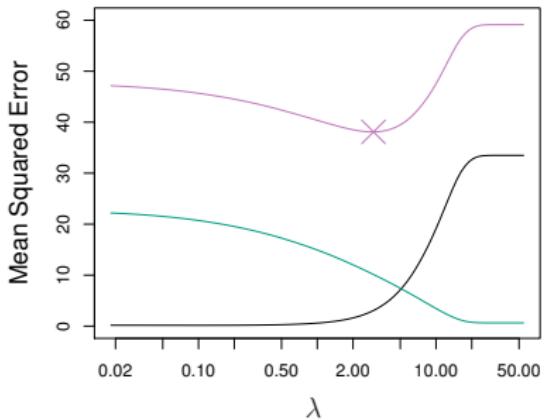


Ridge Regression

$$\bullet : \sum_{j=1}^p \beta_j^2 \leq s$$

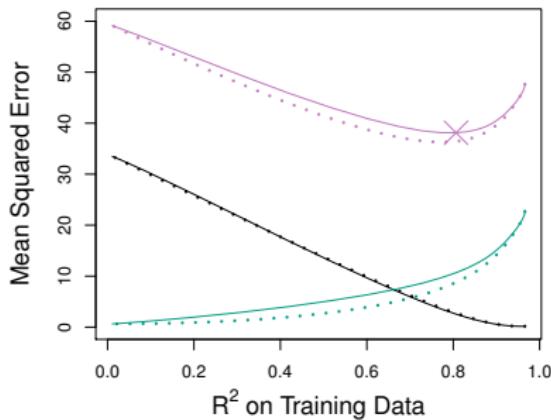
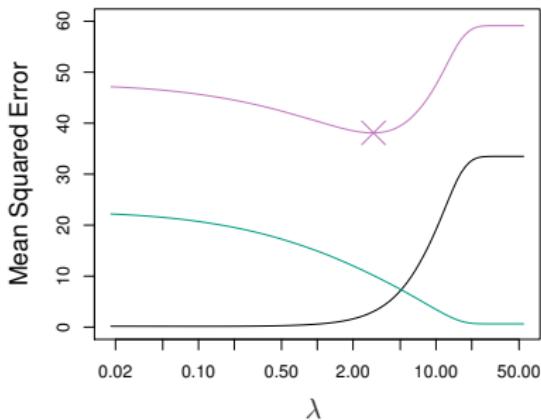
When is the Lasso better than Ridge?

Example 1. Most of the coefficients are non-zero.



When is the Lasso better than Ridge?

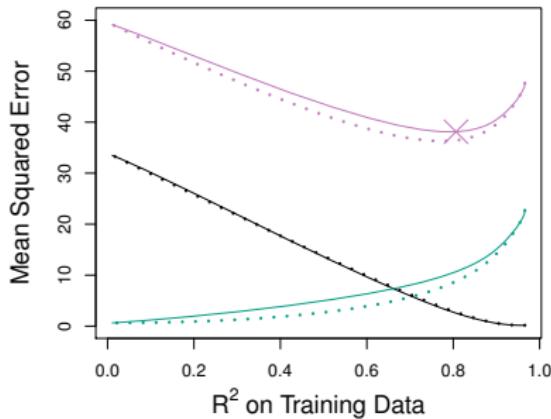
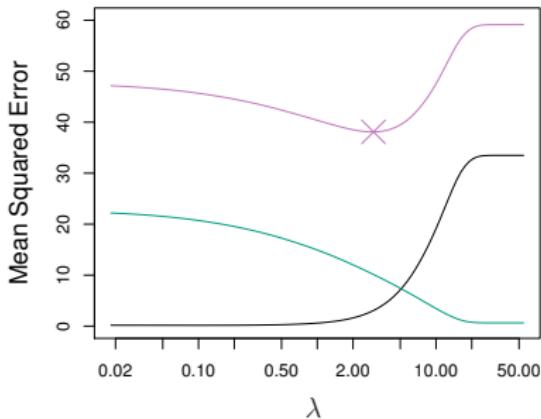
Example 1. Most of the coefficients are non-zero.



- Bias, Variance, MSE.

When is the Lasso better than Ridge?

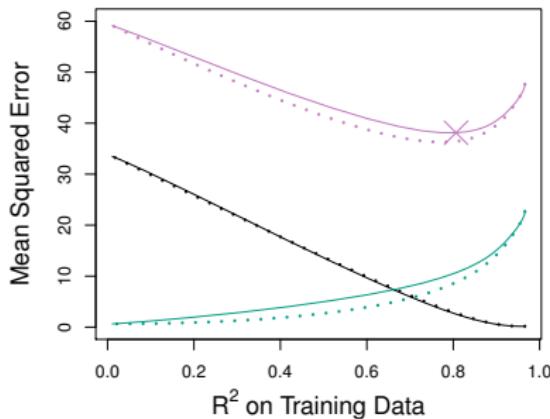
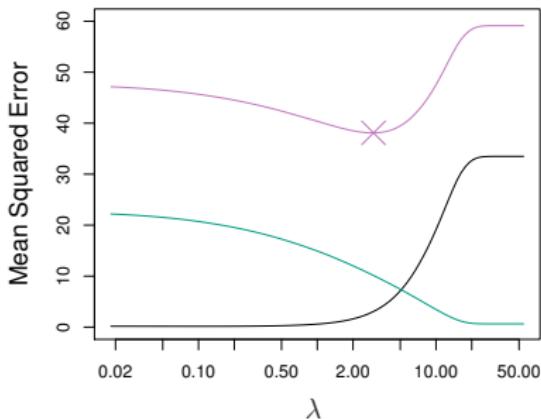
Example 1. Most of the coefficients are **non-zero**.



- Bias, Variance, MSE. The Lasso (—), Ridge (···).

When is the Lasso better than Ridge?

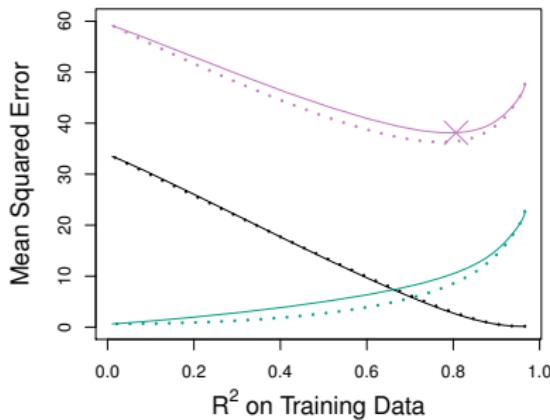
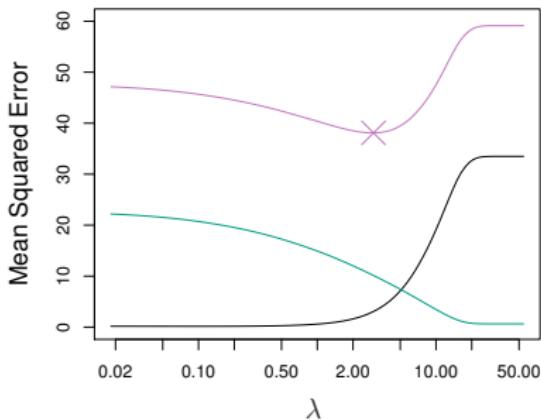
Example 1. Most of the coefficients are non-zero.



- ▶ Bias, Variance, MSE. The Lasso (—), Ridge (···).
- ▶ The bias is about the same for both methods.

When is the Lasso better than Ridge?

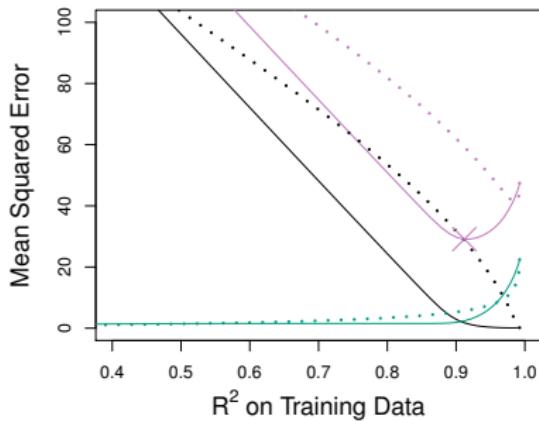
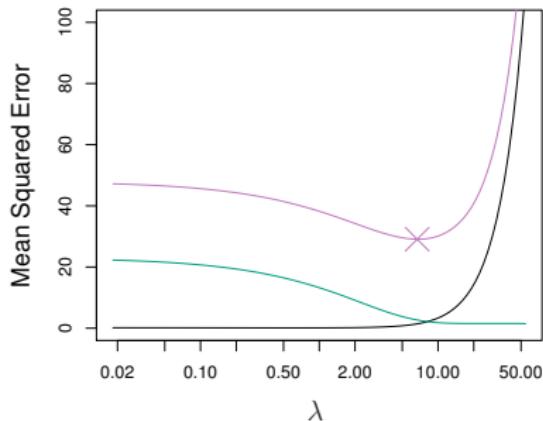
Example 1. Most of the coefficients are non-zero.



- ▶ Bias, Variance, MSE. The Lasso (—), Ridge (···).
- ▶ The bias is about the same for both methods.
- ▶ The variance of Ridge regression is smaller, so is the MSE.

When is the Lasso better than Ridge?

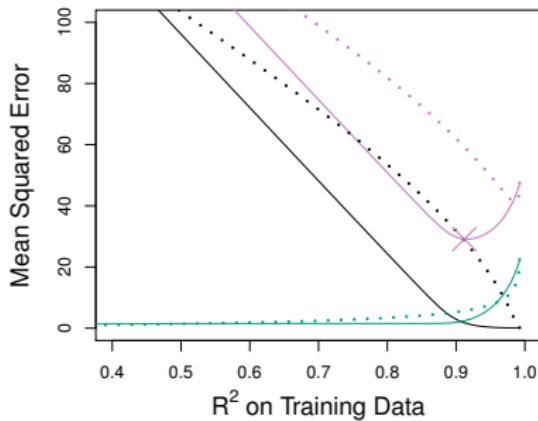
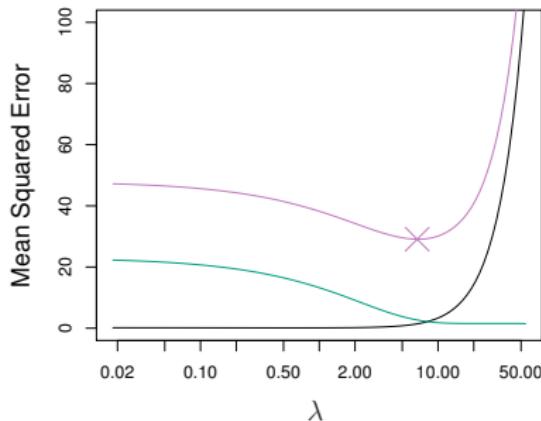
Example 2. Only 2 coefficients are non-zero.



bias is the main difference

When is the Lasso better than Ridge?

Example 2. Only 2 coefficients are non-zero.

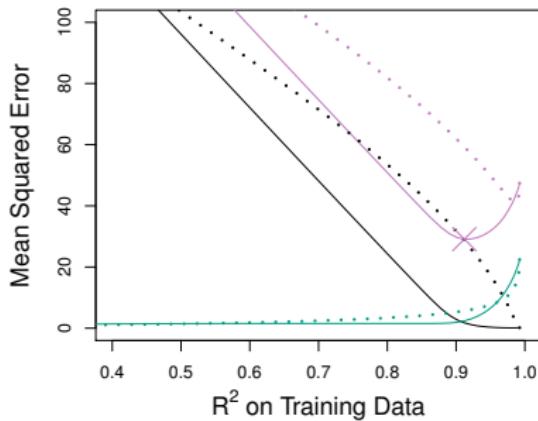
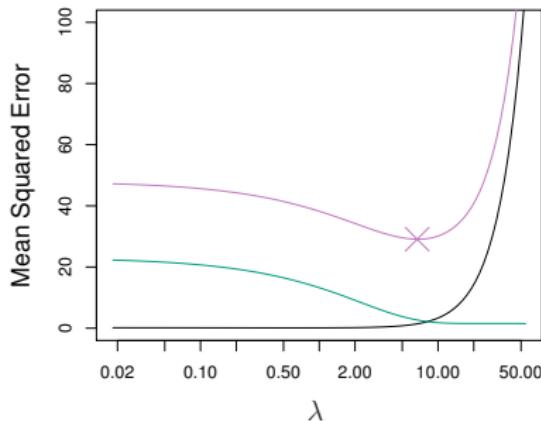


- ▶ Bias, Variance, MSE.

当实际时候的p为零的比较多的时候，我们用Lasso比较好，用ridge会导致比较高bias
但是如果p为零的比较少，用ridge比较好

When is the Lasso better than Ridge?

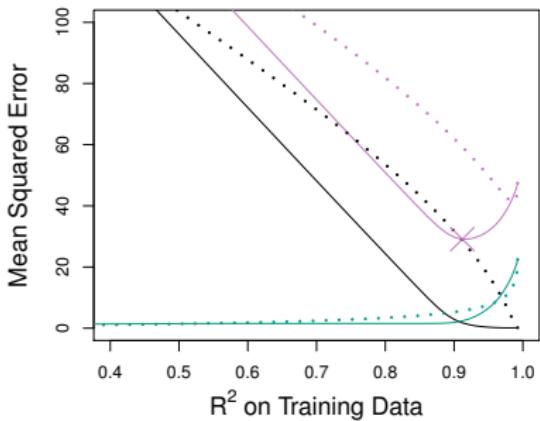
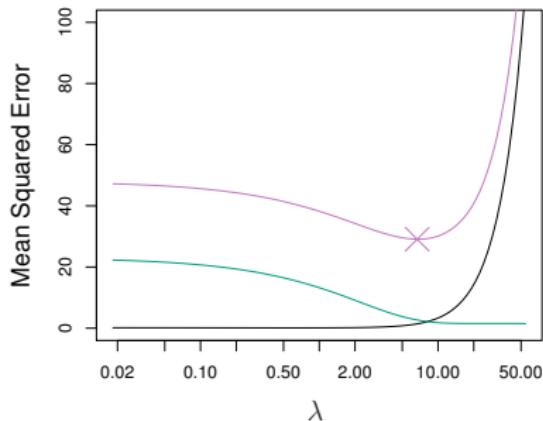
Example 2. Only 2 coefficients are non-zero.



- Bias, Variance, MSE. The Lasso (—), Ridge (···).

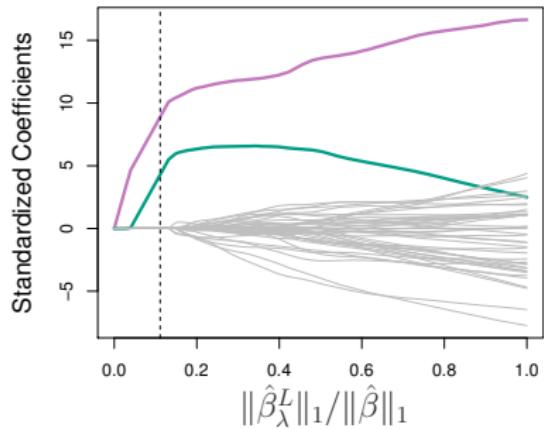
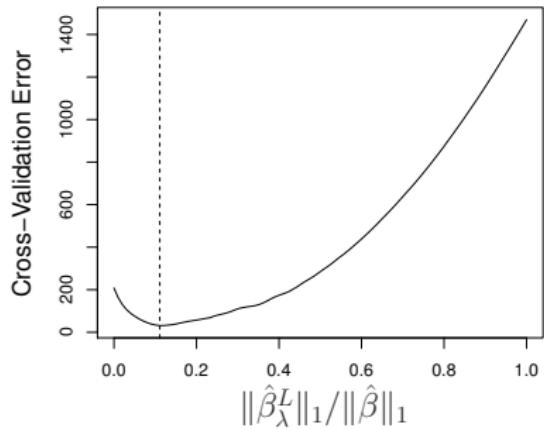
When is the Lasso better than Ridge?

Example 2. Only 2 coefficients are non-zero.



- ▶ Bias, Variance, MSE. The Lasso (—), Ridge (···).
- ▶ The bias, variance, and MSE are lower for the Lasso.

Choosing λ by cross-validation



A very special case

Suppose $n = p$ and our matrix of predictors is $\mathbf{X} = I$.

A very special case

Suppose $n = p$ and our matrix of predictors is $\mathbf{X} = I$.

Then, the objective function in Ridge regression can be simplified:

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

A very special case

Suppose $n = p$ and our matrix of predictors is $\mathbf{X} = I$.

Then, the objective function in Ridge regression can be simplified:

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

and we can minimize the terms that involve each β_j separately:

$$(y_j - \beta_j)^2 + \lambda \beta_j^2.$$

A very special case

Suppose $n = p$ and our matrix of predictors is $\mathbf{X} = I$.

Then, the objective function in Ridge regression can be simplified:

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

and we can minimize the terms that involve each β_j separately:

$$(y_j - \beta_j)^2 + \lambda \beta_j^2.$$

It is easy to show that

$$\hat{\beta}_j^{\text{ridge}} = \frac{y_j}{1 + \lambda}.$$

A very special case

Similar story for the Lasso; the objective function is:

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

A very special case

Similar story for the Lasso; the objective function is:

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

and we can minimize the terms that involve each β_j separately:

$$(y_j - \beta_j)^2 + \lambda |\beta_j|.$$

A very special case

Similar story for the Lasso; the objective function is:

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

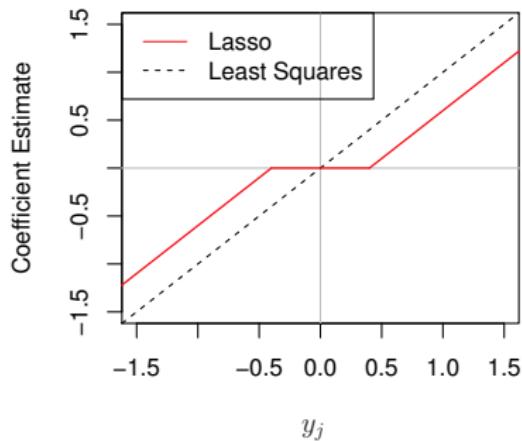
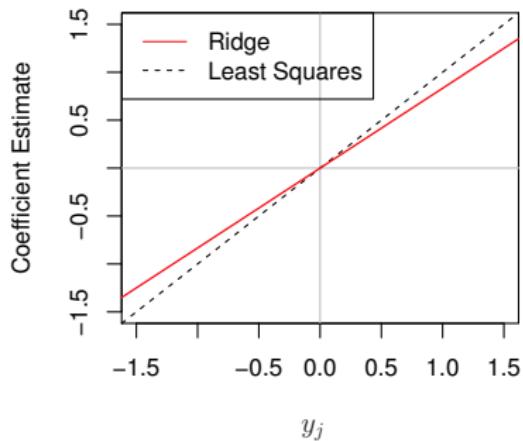
and we can minimize the terms that involve each β_j separately:

$$(y_j - \beta_j)^2 + \lambda |\beta_j|.$$

It is easy to show that

$$\hat{\beta}_j^{\text{lasso}} = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2; \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2; \\ 0 & \text{if } |y_j| < \lambda/2. \end{cases}$$

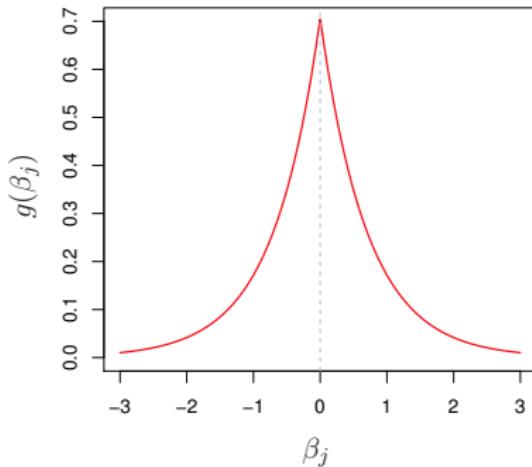
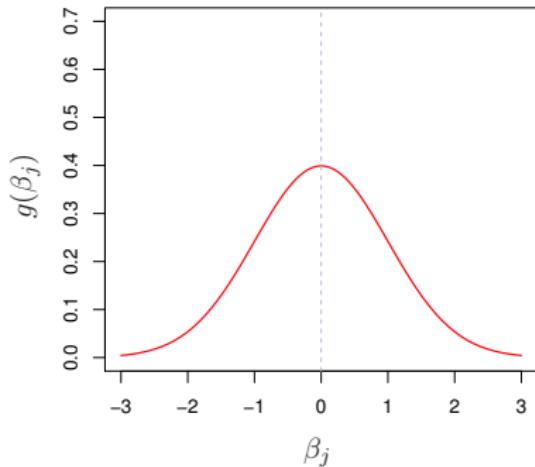
Lasso and Ridge coefficients as a function of λ



Bayesian interpretations

Ridge: $\hat{\beta}^{\text{ridge}}$ is the posterior mean, with a Normal prior on β .

Lasso: $\hat{\beta}^{\text{lasso}}$ is the posterior mode, with a Laplace prior on β .



Summary: Regression

Methods we have discussed:

- ▶ Linear regression with least squares
- ▶ Ridge regression, Lasso

Note: All of these are linear. The solutions are hyperplanes. The different methods differ only in how they *place* the hyperplane.

Summary: Regression

Ridge regression

Suppose we obtain two training samples \mathcal{X}_1 and \mathcal{X}_2 from the same distribution.

- ▶ Ideally, the linear regression solutions on both should be (nearly) identical.
- ▶ With standard linear regression, the problem may not be solvable (if $\mathbf{X}^T \mathbf{X}$ not invertible).
- ▶ Even if it is solvable, if the matrices $\mathbf{X}^T \mathbf{X}$ are close to singular (small spectral condition $c(\mathbf{X}^T \mathbf{X})$), then the two solutions can differ significantly.
- ▶ Ridge regression stabilizes the inversion of $\mathbf{X}^T \mathbf{X}$.

Consequences:

- ▶ Regression solutions for \mathcal{X}_1 and \mathcal{X}_2 will be almost identical if λ sufficiently large.
- ▶ The price we pay is a bias that grows with λ .

Summary: Regression

Lasso

- ▶ The ℓ_1 -constraint "switches off" dimensions; only some of the entries of the solution $\hat{\beta}^{\text{lasso}}$ are non-zero (sparse $\hat{\beta}^{\text{lasso}}$).
- ▶ This variable selection also stabilizes $\mathbf{X}^T \mathbf{X}$, since we are effectively inverting only along those dimensions which provide sufficient information.
- ▶ No closed-form solution; use numerical optimization.

Formulation as optimization problem

Method	$f(\beta)$	Penalty	Solution method
Least squares	$\ \mathbf{y} - \mathbf{X}\beta\ _2^2$	0	Analytic solution exists if $\mathbf{X}^T \mathbf{X}$ invertible
Ridge regression	$\ \mathbf{y} - \mathbf{X}\beta\ _2^2$	$\ \beta\ _2^2$	Analytic solution exists
Lasso	$\ \mathbf{y} - \mathbf{X}\beta\ _2^2$	$\ \beta\ _1$	Numerical optimization