Lecture 6: Bayesian Analysis in Machine Learning

UNIVERSITY OF
SAN FRANCISCO

James D. Wilson

MSAN 628

Computational Statistics

Outline

- Penalized Regression Methods
  - Lasso
  - Ridge Regression
- Classification
  - Bayes Classifier
  - Naïve Bayes
  - Discriminant Analysis

Regression Methods

Recall Linear Regression

Model:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i$$

Estimate:

$$\hat{\beta}_{OLS} = \text{argmin}_\beta \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right)$$

Questions:

1. What if we are primarily concerned with variable selection?
2. What if $p > n$? (high dimensional regression)

Recall: Best Subset Selection

Algorithm

Given: k predictors $\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$

Loop: for (k in 1 to p)

1. Fit all $\binom{p}{k}$ models that contain k predictors
2. Pick M_k = the "best" among these models

Return: $M^* \in \{M_0, \ldots, M_p\}$: $M^* = \text{argmin}_j(S(M_j))$

- $S(M_j)$ = prediction criterion (Mallow's C_p, AIC, BIC, MSPE)

Recall: Best Subset Selection

Important Considerations:

1. Computational Complexity: must fit 2^p models
2. Algorithm is exhaustive: we *will find* the "best" model
3. Often replaced with approximate and less intensive algorithms:
    - Forward stepwise selection
    - Backward stepwise selection
    - Forward-backward stepwise selection

A Nice Alternative: Shrinkage Methods

1. Fits all p predictors using a technique that constrains or regularizes the coefficient estimates by optimizing a slightly different objective function
2. Equivalently, the techniques shrink coefficient estimates to zero
3. Variance of coefficient estimates are reduced as well! Particularly in high dimensional settings!

Ridge Regression

Model:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i$$

Estimate:

$$\hat{\beta}_{Ridge} = \text{argmin}_\beta \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

$\lambda \sum_{j=1}^{p} \beta_j^2$ acts as a shrinkage penalty to standard least squares regression since this value is small when β_j^2 is small.

Note: Also known as Tikhonov regularization

Ridge Regression: regularization perspective

Problem: The variance of $\hat{f}$ for OLS is often high $\Leftrightarrow$ predictions significantly change with small changes in X.

Reason: $X^T X$ is ill-conditioned $\Leftrightarrow$ either $p \approx n$ or variables suffer from multicollinearity:

$$\hat{\beta}_{OLS} = (X^T X)^{-1} X^T y$$

Solution: Ridge regression regularizes $(X^T X)$:

$$\hat{\beta}_{Ridge} = (X^T X + \lambda I)^{-1} X^T y$$

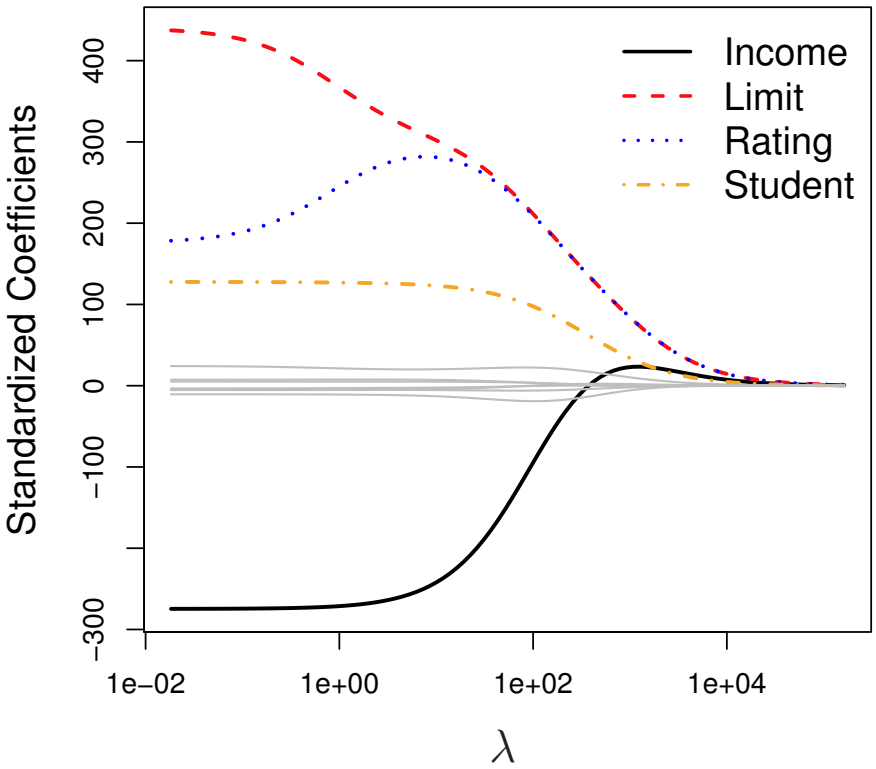
Ridge Regression

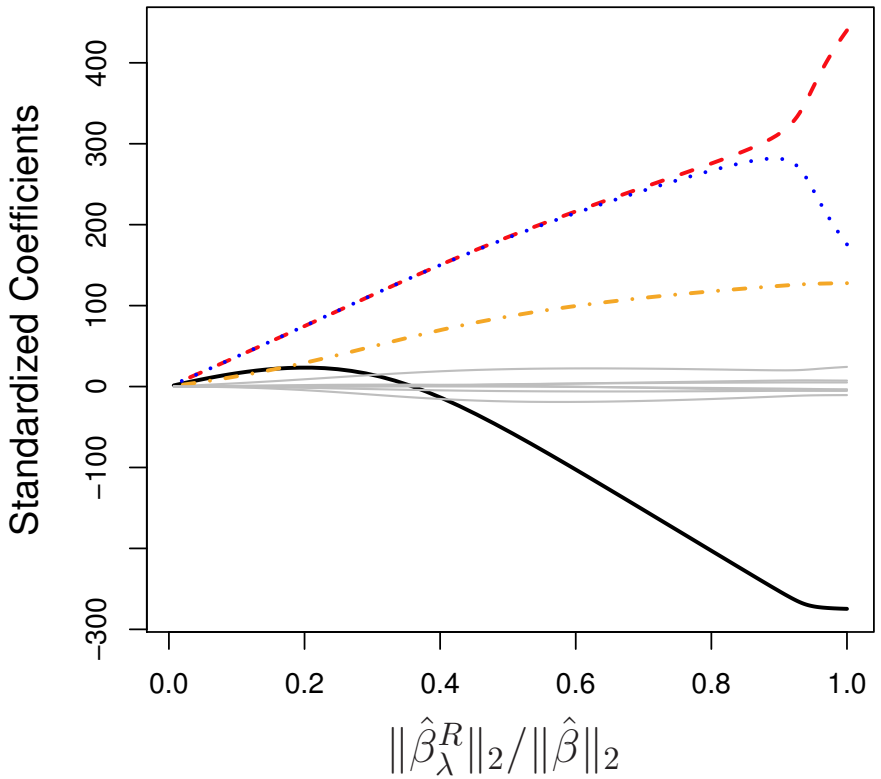
$$\hat{\beta}_{Ridge} = \text{argmin}_{\beta} \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

λ: tuning parameter that adjusts the effect of the penalty

- $\lambda = 0 \quad \Rightarrow \quad \hat{\beta}_{OLS} = \hat{\beta}_{Ridge}$
- $\lambda \to \infty \quad \Rightarrow \quad \hat{\beta}_{Ridge} \to 0$
- λ chosen using cross validation: amazingly can computationally be determined for all possible values simultaneously!

Example: Comparison of $\hat{\beta}_{OLS}$ and $\hat{\beta}_{Ridge}$





When does Ridge Regression outperform OLS?

1. **Computationally**: Ridge estimates for all values of λ can be determined simultaneously with one fit. Significant advantage over best subset selection that requires 2^p least squares fits.

2. **Model Accuracy**: OLS estimates often have high variance but low bias. Increases in λ lead to shrinkage, which subsequently leads to a major decrease in variance and only a slight increase in bias.

3. Key is to look across a grid of λ for best MSPE.

When does Ridge Regression outperform OLS?

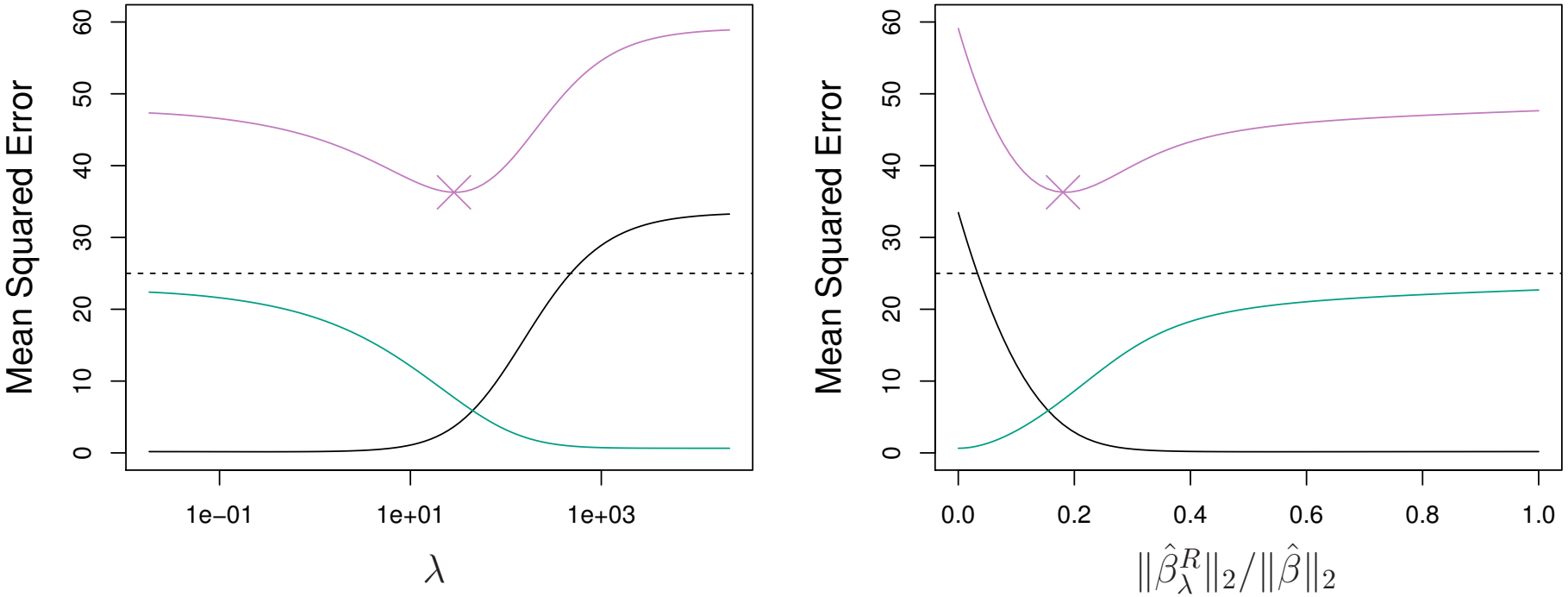


Figure: Squared bias (black), variance (green), and MSPE (purple) for $\hat{\beta}_{Ridge}$ on a simulated data set.

Weaknesses of Ridge Regression

- Requires a user - specified tuning parameter λ
- Interpretability of $\hat{\beta}_{Ridge}$
- **Subtle but important point**: The penalty $\lambda \sum_{j=1}^{p} \beta_j^2$ shrinks β *towards* 0 but does not set any values *exactly* to 0.
  - **Exception**: $\lambda = \infty$ – here all β_j are exactly 0
  - **Consequence**: The saturated model is *always* chosen!

Question: Can we shrink some coefficients exactly to zero?

The Lasso

Least absolute shrinkage and selection operator

Model:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i$$

Estimate:

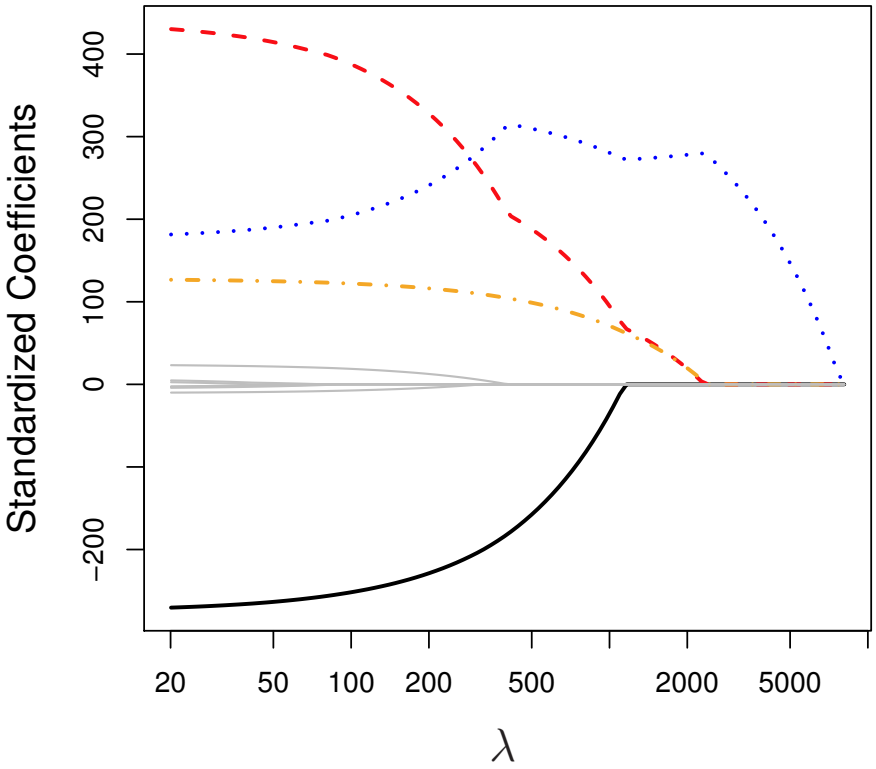
$$\hat{\beta}_{Lasso} = \text{argmin}_{\beta} \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

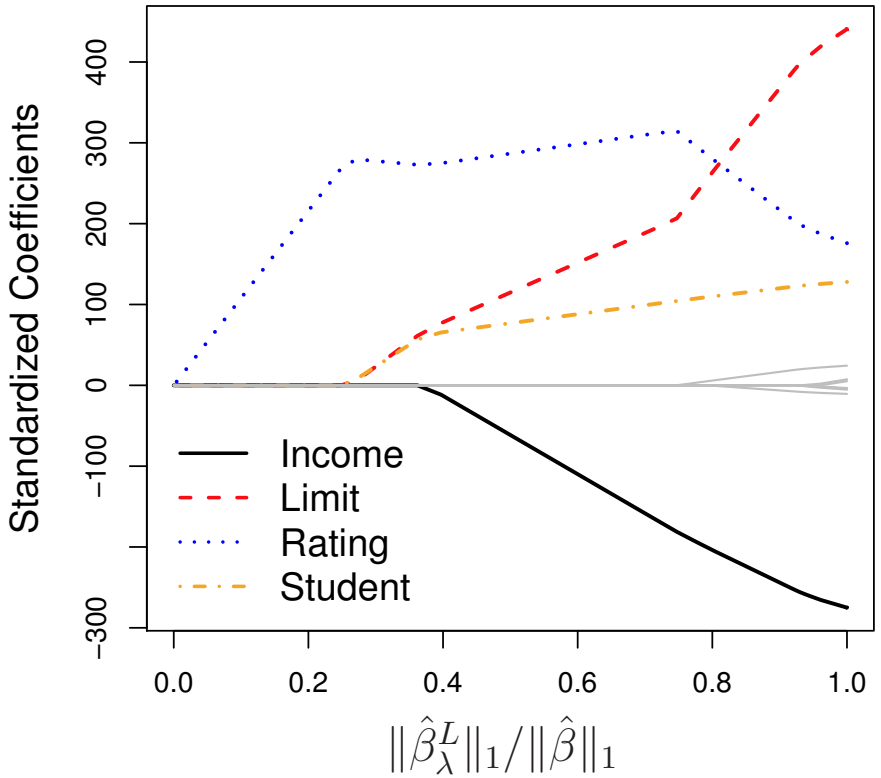
$\lambda \sum_{j=1}^{p} |\beta_j|$ acts as a shrinkage penalty to standard least squares regression since this value is small when $|\beta_j|$ is small.

Historical Note on the Lasso

- From the paper "Regression shrinkage via the lasso" (1996) in *Journal of the Royal Statistical Society. Series B* by Robert Tibshirani (one of the authors of ISL and ESL)
- Considered by many to be the most influential modern statistical method
- Paper currently has 14243 citations! (as of October 27, 2015)
- Website:
  `http://statweb.stanford.edu/ tibs/lasso.html`

Variable Selection Property of Lasso





Note: Changing λ sets various subsets of β to 0! Why?

Re-formulations of Ridge and the Lasso

Both methods can be viewed as optimization problems.

- **Ridge Regression**:

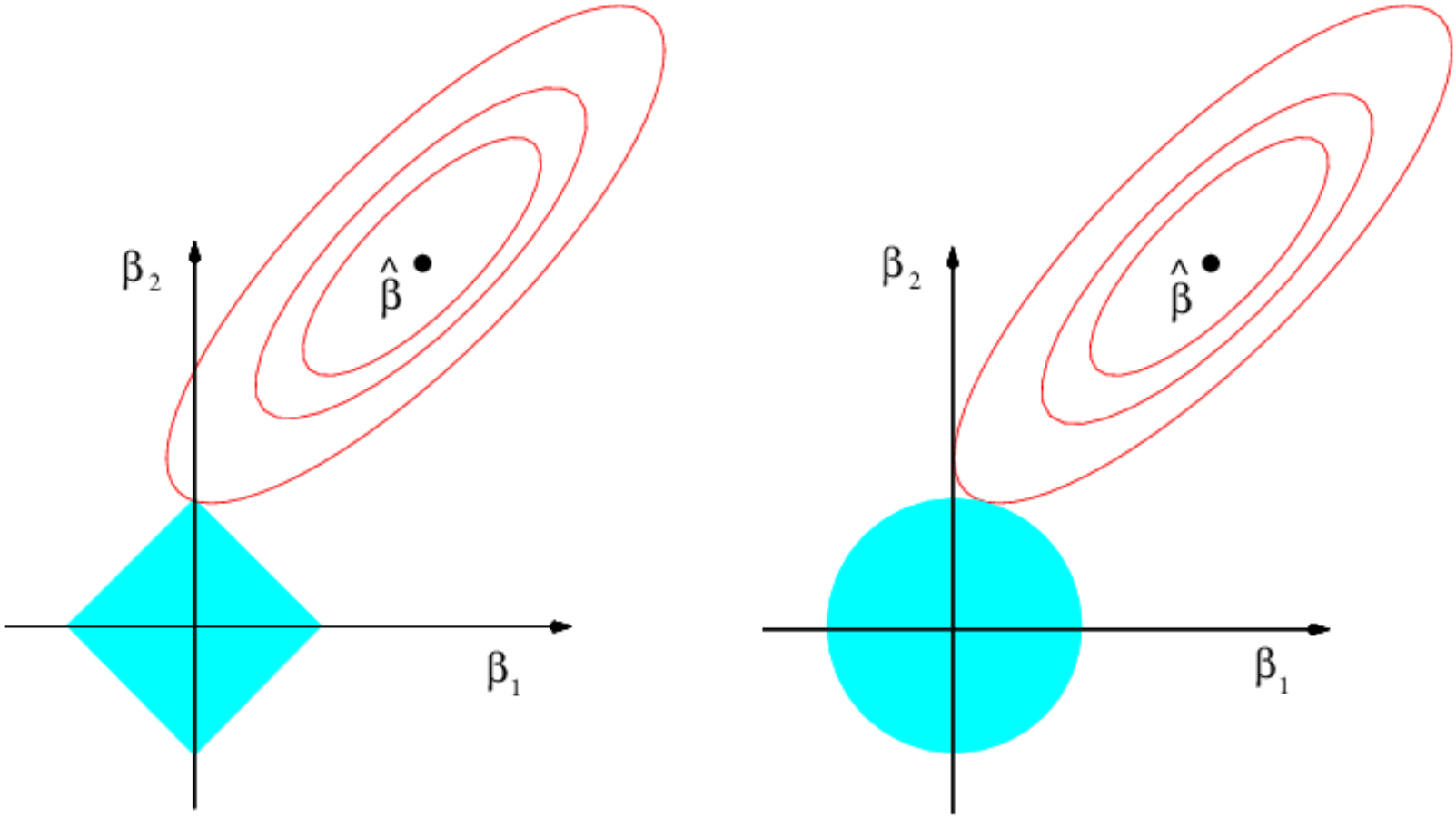
$$\text{minimize}_\beta \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right) \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s$$

- **Lasso**:

$$\text{minimize}_\beta \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right) \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

Uh, ok so what? Explains the variable selection property of the Lasso!

Comparison of Lasso and Ridge



Often, the Lasso shrinks coefficients *exactly* to zero!

Comparison of Lasso and Ridge

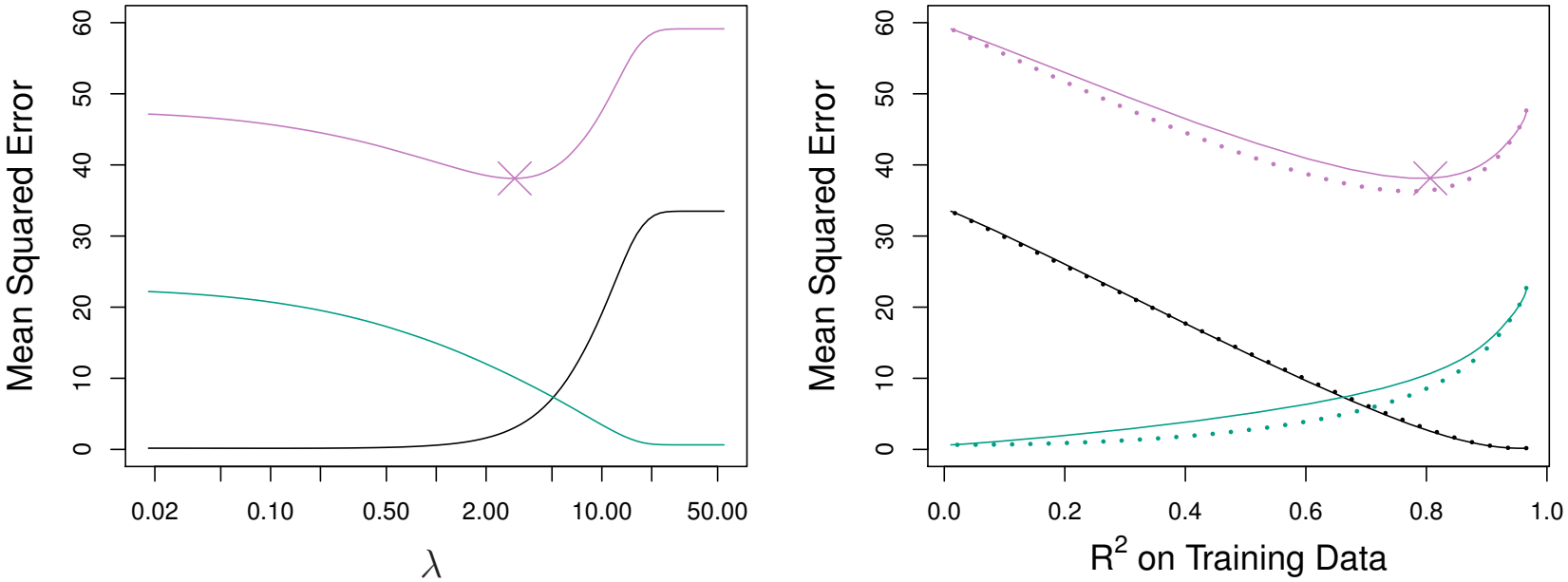


Figure: Squared bias (black), variance (green), and MSPE (purple). Dashed = Ridge, solid = Lasso

Note: Simulated data here included 45 / 45 non-zero coefficients. So, *no* variable selection is needed.

Comparison of Lasso and Ridge

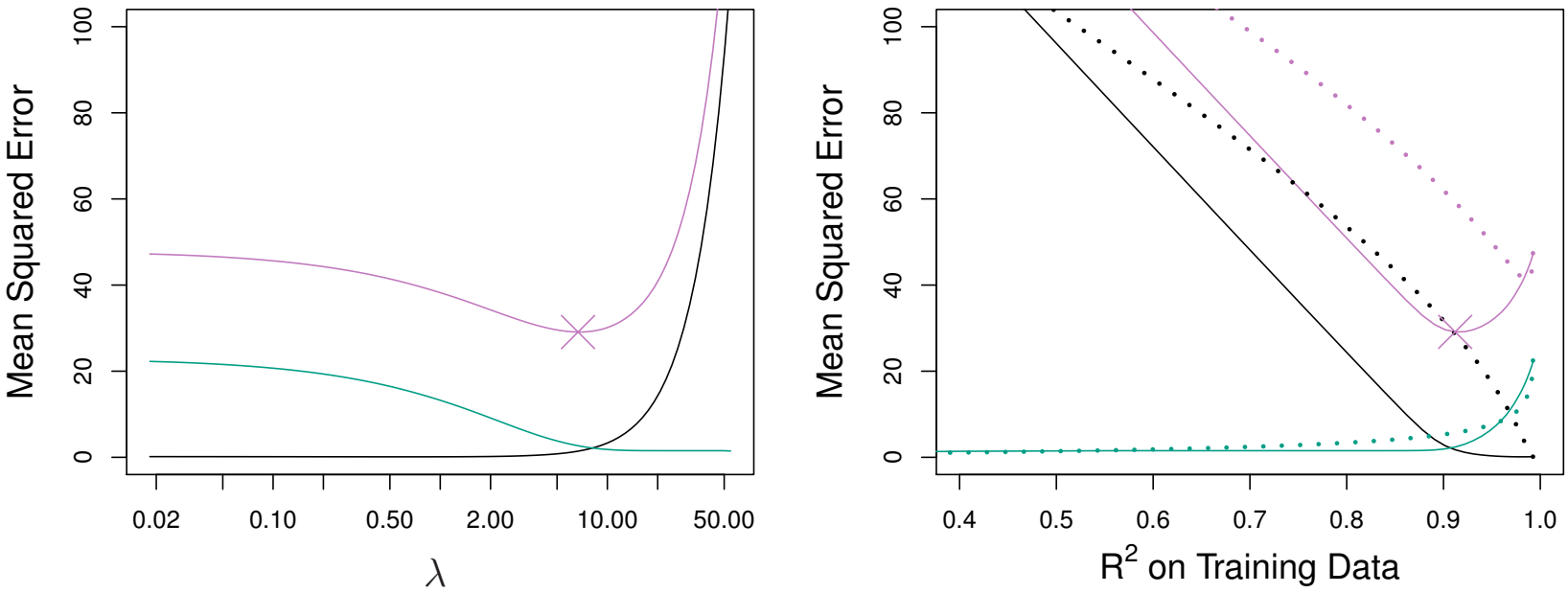


Figure: Squared bias (black), variance (green), and MSPE (purple). Dashed = Ridge, solid = Lasso

Note: Simulated data here included 2 / 45 non-zero coefficients. So, variable selection *is* needed.

Elastic Net

In general, the Lasso is best for variable selection / sparse relationships; Ridge for ill-conditioned problems.

Elastic Net: Combines Lasso and Ridge

Model:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i$$

Estimate:

$$\hat{\beta}_{EN} = \text{argmin}_\beta \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \alpha\lambda \sum_{j=1}^{p} |\beta_j| + \frac{1-\alpha}{2} \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

Best of both worlds? – well, this is more difficult to interpret!

Elastic Net Properties

1. $\alpha = 0$: reduces to Ridge Regression
2. $\alpha = 1$: reduces to Lasso
3. Has both properties of Ridge and Lasso:
    1. Reduces variance
    2. Variable selection
4. Recently proven that Elastic Net is equivalent to linear support vector machines.

A Bayesian Perspective of Ridge and Lasso

Let $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$ be the parameters in a linear regression.

Bayesian Framework: Assume that β is a *random vector* with distribution $p(\beta)$. Here,

- $f(y|X, \beta)$ = likelihood of the data (Gaussian if ϵ is Gaussian)
- $\pi(\beta)$ = prior distribution of β
- $p(\beta|X, y)$ =posterior distrubtion of β given (X, y)

A Bayesian Perspective of Ridge and Lasso

Assumption 1: $\pi(\beta) = \prod_{i=1}^{p} g(\beta_i)$ (i.e. β_i's are iid).

Under Assumption 1, the regression model becomes:

$$y = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p + \epsilon$$

$$\beta_i \overset{iid}{\sim} g(x)$$

A Bayesian Perspective: Ridge Estimators

Let $h(\lambda)$ be some positive monotonic function of λ

Ridge Coefficients

- **Data**: $y \mid (\beta, X) \sim N(X\beta, \sigma^2 I_n)$ (Linear regression)
- **Prior**: $\beta_j \overset{iid}{\sim} N(0, h(\lambda))$

Fact: Under this specification,

$$\hat{\beta}_{Ridge} = \text{Mode}(p(\beta|X, y))$$

A Bayesian Perspective: Lasso Estimators

Let $h(\lambda)$ be some positive monotonic function of λ

Lasso Coefficients

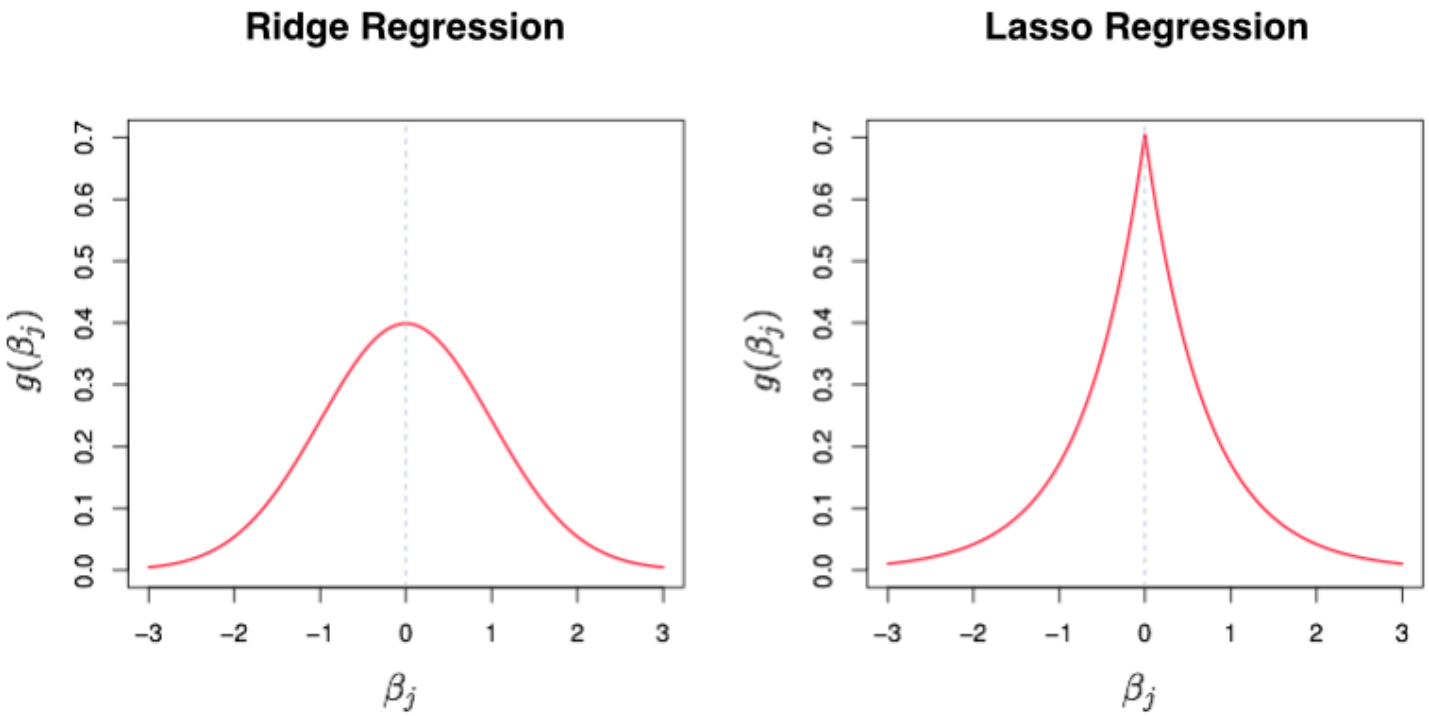
- **Data**: $y \mid (\beta, X) \sim N(X\beta, \sigma^2 I_n)$ (Linear regression)
- **Prior**: $\beta_j \overset{iid}{\sim}$ Laplace distribution (double exponential) with mean 0 and variance $h(\lambda)$

Fact: Under this specification,

$$\hat{\beta}_{Lasso} = \text{Mode}(p(\beta|X, y))$$

Conclusion: Under appropriate prior specifications, the Ridge and Lasso estimates are the maximum a posteriori (MAP) estimators for β.

Prior Distributions



Another way of understanding the likelihood of shrinkage!

Selecting λ

General Method: Grid search and cross-validation

1. Fix a value of λ
2. Estimate model and calculate average MSPE from k-fold cross-validation
3. Repeat the above procedure across a grid of λ
4. Choose λ that leads to smallest MSPE

Important: The above procedure can be done in parallel, easing computation.

Implementing Shrinkage Methods with R

Now we show how to implement the Lasso, Ridge Regression and Elastic Net in R.

Go to the *Shrinkage.Rmd* document in the *Files / Code* folder on Canvas.

Classification

Recall: Classification Setting

Training Data: Consisting of n observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

- y_i are discrete valued observations

Test Data: Observations of the form $(\mathbf{x}_o, y_o)$.

Goal:

- Train a classifier $\phi(x) = \hat{y}$ using the training data.
- Identify the classifier that minimizes the MSPE on the test data:

$$\text{Ave}(\mathbb{I}(y_o \neq \hat{y}_o))$$

The Bayes Classifier

Theorem

Minimizing Ave($\mathbb{I}(y_o \neq \hat{y}_o)$), on average, is equivalent to choosing the class j for which the quantity

$$\mathbb{P}(Y = j \mid X = \mathbf{x}_o)$$

is largest.

The classifier $\phi(\mathbf{x}_o) = \text{argmax}_j\,(\mathbb{P}(Y = j \mid X = \mathbf{x}_o))$ is the Bayes Classifier.

Key Question: How do we calculate the Bayes Classifier, and what exactly do we mean by this *conditional probability*?

Stochastic Setting (Binary case)

Regard observations $(X_1, Y_1), \ldots, (X_n, Y_n)$ as being independent samples from a fixed distribution $\mathbb{P}$ on $\mathcal{X} \times \{-1, +1\}$

Notation: use (X, Y) to denote a generic pair with distribution $\mathbb{P}$ and independent of the observations.

Quantities of Interest (Bayesian statistics revisited...)

1. Prior probabilities of $Y = +1$ and $Y = -1$
2. Conditional probability of $Y = +1$ given $X = \mathbf{x}$
3. Class conditional distributions of X given $Y = y$

Prior Probabilities of Y (Binary case)

Let $\pi_{-1} = \mathbb{P}(Y = -1)$ and $\pi_1 = \mathbb{P}(Y = +1)$

- Probability of seeing class $Y = -1$ or $Y = +1$ *before* (prior to) observing $\mathbf{x}$
- Relative abundance of class -1 and $+1$
- Note $\pi_{-1} + \pi_1 = 1$
- Cases in which $\pi_{-1} >> \pi_1$ or v.v. can be problematic (problem of unbalanced data)

Unconditional and Conditional Densities of X

Assume: $\mathcal{X} \subseteq \mathbb{R}^p$ and X has unconditional joint density $f(\mathbf{x})$:

$$\mathbb{P}(X \in A) \;=\; \int_A f(\mathbf{x})\, d\mathbf{x}, \quad A \subseteq \mathcal{X}.$$

Let $f_y(\mathbf{x})$ denote class-conditional density of X given $Y = y$.

$$\mathbb{P}(X \in A \mid Y = y) \;=\; \int_A f_y(\mathbf{x})\, d\mathbf{x}, \quad A \subseteq \mathcal{X}.$$

Take-away: Class-conditional densities f_{-1} and f_1 tell us about separability of the classes -1s and $+1$s.

Conditional Distribution of Y Given X (Binary case)

Conditional probability of Y given $X = \mathbf{x}$:

$$\eta(\mathbf{x}) = \mathbb{P}(Y = +1 \mid X = \mathbf{x})$$

$$= \text{probability of seeing class } Y = +1 \text{ after observing } \mathbf{x}$$

Note: $\mathbb{P}(Y = -1 \mid X = \mathbf{x}) = 1 - \eta(\mathbf{x})$.

Regimes:

- $\eta(\mathbf{x}) \approx 1 \Rightarrow Y$ is likely to be $+1$
- $\eta(\mathbf{x}) \approx 0 \Rightarrow Y$ is likely to be -1
- $\eta(\mathbf{x}) \approx 1/2 \Rightarrow$ value of Y uncertain

The Bayes Classifier

For binary classification, the Bayes classifier for new data x_o is:

$$\hat{y}_o = \begin{cases} -1 & \text{if} \quad \eta(x) < 0.5 \\ +1 & \text{if} \quad \eta(x) > 0.5 \end{cases}$$

Mathematical Fact: The Bayes classifier $\hat{y}_o$ (for general multi-class classification) has the smallest possible test error rate. This error is called the Bayes error rate and is given by:

$$1 - \mathbb{E}[\max_j \{\mathbb{P}(Y = j \mid X)\}]$$

This value is analagous to the *irreducible error* in regression.

So, the bayes classifier is the best that we can hope to obtain, but...

Bayes Theorem: relationship among distributions

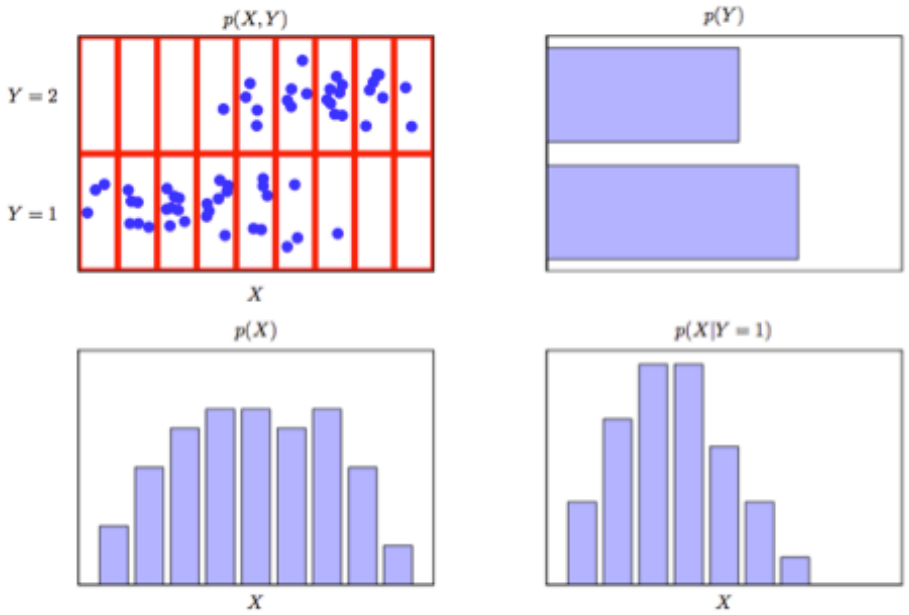
Bayes Theorem gives the following relationship:

$$\mathbb{P}(Y = j \mid X = \mathbf{x}) = \frac{\pi_j f_j(\mathbf{x})}{f(\mathbf{x})} = \frac{\pi_j f_j(\mathbf{x})}{\sum_{j=1}^{m} \pi_j f_j(\mathbf{x})}$$

Key (and unfortunate) point: To obtain the bayes classifier, we need

- Class conditional probabilities: $f_j(\mathbf{x}), j = 1, \ldots, m$
- Prior probabilities $\pi_j, j = 1, \ldots, m$

How do we estimate probabilities?



Two major choices:

- Make assumptions about data. Example: (X, Y) are iid from some distribution
- Empirical estimation of joint density of (X, Y) (i.e. histogram approach)

Summary of Bayes Classifier

- If we knew the class conditional probabilities of X given $Y = y$ and the prior probabilities assoicated with Y, then the Bayes classifier is the best we can do in classification.
- In some applications, it is reasonable to model the above densities based on prior knowledge and statistical inference (e.g., multivariate Guassian for $f_j(\mathbf{x})$)
- In the applications that we cannot provide a model, we have to *estimate* these probabilities.
  - Easily done for π_j:

$$\hat{\pi}_j = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(y_i = j)$$

  - The joint pdf of $f_j(\mathbf{x})$ is challenging without further assumptions...

Naïve Bayes Classifiers

Recall: $f_j(\mathbf{x}) = f(\mathbf{x} \mid Y = j)$

Major (simplifying) Assumption: given Y, features / predictors are conditionally independent of one another:

$$f_j(\mathbf{x}) = \prod_{k=1}^{p} f(x_k \mid Y = j)$$

Result: $f(x_k \mid Y = j)$ can be easily estimated via an empirical (histogram) approach. This is significantly easier than estimating the full joint density of $f_j(\mathbf{x})$

Naïve Bayes Classifiers

Algorithm

Given: Training observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$, test observation $\mathbf{x}_o$

Estimate:

- $f(x_k \mid Y = j)$ for all **variables** $k = 1, \ldots, p$, and **classes** $j = 1, \ldots, m$
- π_j and $f_j(\mathbf{x}) = \prod_{k=1}^{p} f(x_k \mid Y = j)$ for all j

Calculate:

$$\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x}_o) = \frac{\hat{\pi}_j \hat{f}_j(\mathbf{x}_o)}{\sum_{j=1}^{m} \hat{\pi}_j \hat{f}_j(\mathbf{x}_o)}, \qquad j = 1, \ldots, m$$

Return: Classifier $\phi(\mathbf{x}_o)$ where

$$\phi(\mathbf{x}_o) = \text{argmax}_j(\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x}_o))$$

Event Models for Naïve Bayes

In some cases, it is reasonable to model the class conditional distributions using well-established probabilistic models (think back to your favorite probability course).

For example, consider cases where $X \mid Y = y$ is

- Continuous $\rightarrow$ Gaussian RV
- Count the occurrence of each feature $\rightarrow$ Multinomial RV
- Observation of a feature as a binary variable $\rightarrow$ Bernoulli RV

Example: Gaussian Naïve Bayes

- Assume the likelihood of the features is Gaussian
- Use a parametric likelihood function of real-valued variable X

$$f_i(x) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$

where $\mu_j := \mathbb{E}[X \mid Y = j]$ is the conditional mean and $\sigma_j^2 := \text{Var}(X \mid Y = j)$ is the conditional variance of X given $Y = j$

- The posterior probability is evaluated as a product of univariate conditional density functions

$$\mathbb{P}(Y = j \mid X = \mathbf{x}) \propto \pi_j \prod_{i=1}^{p} f_i(x)$$

Example: Multinomial Naïve Bayes

- X vectors represent the frequencies with which certain events (one per feature) have been generated by a multinomial $(p_1, p_2, \ldots, p_p)$

Example: Probabilities of words appearing in documents

- Documents represented as counts for words that appear in it
- Independence assumption is that the presence of a word is conditionally independent of the presence of another one, given y

Example: Bernoulli Naïve Bayes

Longer name: multivariate Bernoulli

- X vectors are binary variables

Example: $Y = 1$ if a word appears

- Document represented as binary feature vector
- Independence assumption means the presence of a word is conditionally independent of the presence of another one, given Y

Example: Empirical Naïve Bayes

Example: Spam in the Enron Email Corpus

You'd like to develop a spam filter based on the words in the Enron emails from the Enron email directory in 2001. These emails have already been filtered into `spam` emails and `normal` emails. In particular, you'd like to build a filter based on if the word "meeting" is in a new email.

Data is available at `https://www.cs.cmu.edu/~enron/`.

Example: Spam Filter for Individual Words

Digging into the data, you calculate the following empirical probabilities:

- $\hat{\mathbb{P}}(\texttt{spam}) = 0.29$
- $\hat{\mathbb{P}}(\texttt{normal}) = 0.71$
- $\hat{\mathbb{P}}(\text{"meeting"} \mid \texttt{spam}) = 0.0106$
- $\hat{\mathbb{P}}(\text{"meeting"} \mid \texttt{normal}) = 0.0416$

Thus, we can directly obtain:

$$\hat{\mathbb{P}}(\texttt{spam} \mid \text{"meeting"}) = \frac{0.0106 * 0.29}{(0.0106 * 0.29 + 0.0416 * 0.71)} = 0.09 = 9\%$$

Naïve Bayes Classifier Review

- Approximation of the Bayes Classifier
- Assumes that $X_i \mid Y = y$, $i = 1, \ldots, n$ are independent
- Easy to implement
- Requires a choice of models for the prior distribution of Y and the class-conditional distribtuion of X given $Y = y$.
- Requires thresholding to determine classification

Recall: Bayes Classifiers

Training Data: n observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ with $y_i \in \{1, \ldots, m\}$

Test Data: Observations of the form $(\mathbf{x}_o, y_o)$.

Bayes Theorem gives the following relationship:

$$\mathbb{P}(Y = j \mid X = \mathbf{x}) = \frac{\pi_j f_j(\mathbf{x})}{f(\mathbf{x})} = \frac{\pi_j f_j(\mathbf{x})}{\sum_{j=1}^{m} \pi_j f_j(\mathbf{x})}$$

To calculate the Bayes classifier, we need

- Class conditional probabilities: $f_j(\mathbf{x})$ (*difficult!*)
- Prior probabilities π_j (*pretty easy*)

Discriminants

Definition

In many cases, we can view a classifier $\phi(\mathbf{x})$ as an optimization of some function of $\mathbf{x}$. Namely,

$$\phi(\mathbf{x}) = \text{argmax}_j \left(\delta_j(\mathbf{x}) \right)$$

The function $\delta_j(\mathbf{x})$ is the discriminant of $\mathbf{x}$ as it is used to *discriminate* between classes of Y. Note that $\delta_j(\mathbf{x})$ is also the decision region for class $j \in \{1, \ldots, m\}$.

Example: For the Bayes classifier,

$$\delta_j(\mathbf{x}) = \mathbb{P}(Y = j \mid X = \mathbf{x})$$

Linear and Quadratic Discriminants

- The Bayes classifier discriminant need not take a simple form.
- We can talk about special (simple) cases of Bayes classifiers.
- We will talk about two classes of discriminants:
  - Linear Discriminants: $\delta_j(\mathbf{x})$ is a *linear* function of $\mathbf{x}$. For some matrices $\{A_j\}$ and vectors $\{\mathbf{b}_j\}$,

$$\delta_j(\mathbf{x}) = \mathbf{x}^T A_j + \mathbf{b}_j$$

  - Quadratic Discriminants: $\delta_j(\mathbf{x})$ is a *quadratic* function of $\mathbf{x}$. For some matrices $\{A_j\}$, $\{B_j\}$ and vectors $\{\mathbf{b}_j\}$,

$$\delta_j(\mathbf{x}) = \mathbf{x}^T A_j \mathbf{x} + \mathbf{x}^T B_j + \mathbf{b}_j$$

Linear and Quadratic Discriminants

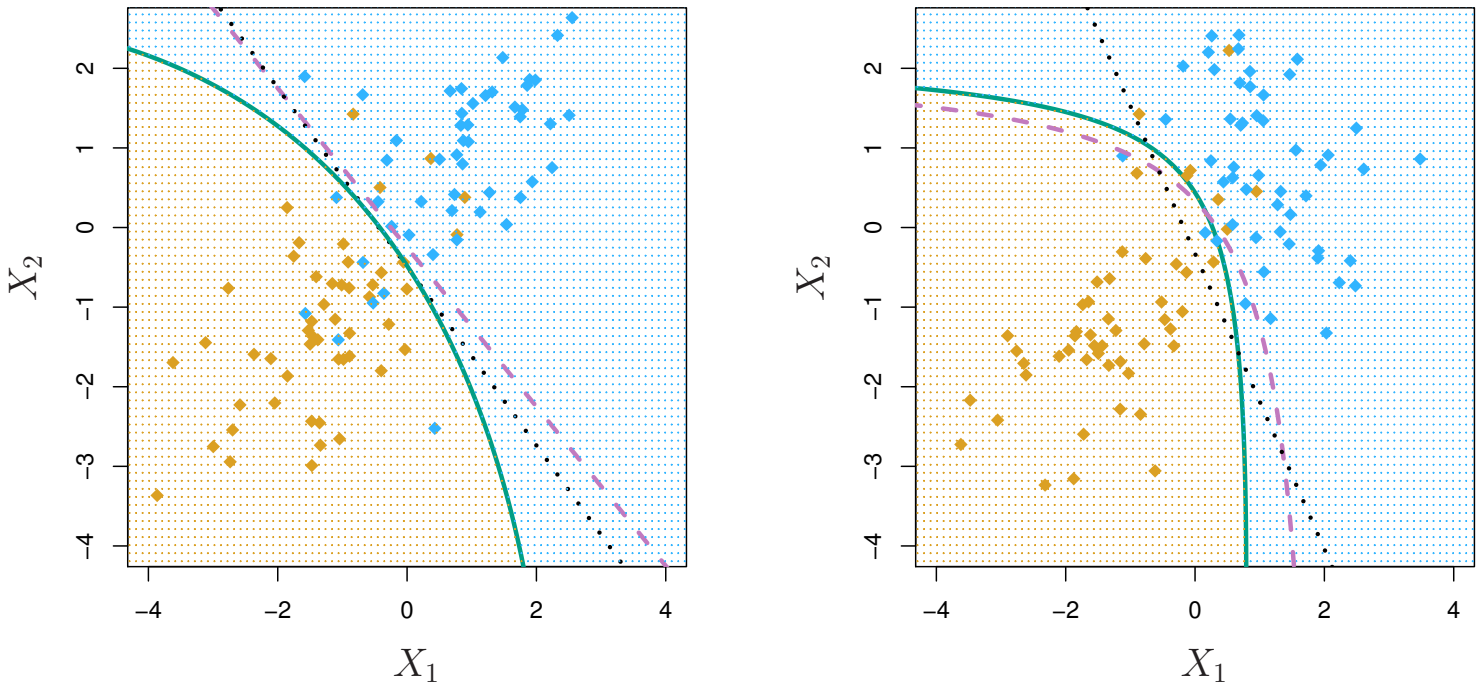


Figure: Linear vs. Quadratic Discriminants. The purple dashed line represents the true Bayes classifier.

Linear Discriminants: $p = 1$ case

- Suppose that there is only one predictor ($p = 1$) and Y takes on a class $j \in \{1, \ldots, m\}$
- **Distributional Assumption**: $X \mid Y = j$ is Guassian with mean μ_j and the *same* variance σ^2:

$$f_j(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_j)^2\right)$$

- Then applying Bayes theorem and doing some algebra gives

$$\log\left(\mathbb{P}(Y = j \mid X = x)\right) = x * \frac{\mu_j}{\sigma^2} - \frac{\mu_j^2}{2\sigma^2} + \log(\pi_j)$$

Linear Discriminants: $p = 1$ case

Fact: Let $f(x) \geq 0$ for all x. Then maximizing $f(x)$ is equivalent to maximizing the function $g(x) = \log(f(x))$. (why?...)

Conclusion: If we assume that $X \mid Y = j$ as $N(\mu_j, \sigma^2)$, we can derive the discriminant function:

$$\delta_j(x) = x * \frac{\mu_j}{\sigma^2} - \frac{\mu_j^2}{2\sigma^2} + \log(\pi_j)$$

Question: We don't know μ_j and σ^2. How can we estimate them?

Linear Discriminants: $p = 1$ case

Let $Y \in \{1, \ldots, m\}$. Then we can estimate μ_j and σ^2 using

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n} x_i \, \mathbb{I}(y_i = j)$$

$$\hat{\sigma}^2 = \frac{1}{n - m} \sum_{j=1}^{m} \sum_{i=1}^{n} (x_i - \hat{\mu}_j)^2 \, \mathbb{I}(y_i = j)$$

As usual, we can estimate π_j using:

$$\hat{\pi}_j = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(y_i = j)$$

Linear Discriminants: $p = 1$ case

The linear discriminants for Y are given by:

$$\hat{\delta}_j(x) = x * \frac{\hat{\mu}_j}{\hat{\sigma}^2} - \frac{\hat{\mu}_j^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_j), \qquad j \in \{1, \ldots, m\}$$

In the simple case that $m = 2$, we can show that the Bayes decision boundary corresponds to the point where

$$x = \frac{\hat{\mu}_1^2 - \hat{\mu}_2^2}{2(\hat{\mu}_1 - \hat{\mu}_2)} = \frac{\hat{\mu}_1 + \hat{\mu}_2}{2}$$

Note: The Bayes decision boundary above is exactly the point where $\hat{\delta}_{-1}(x) = \hat{\delta}_{+1}(x)$

Linear Discriminants: Example

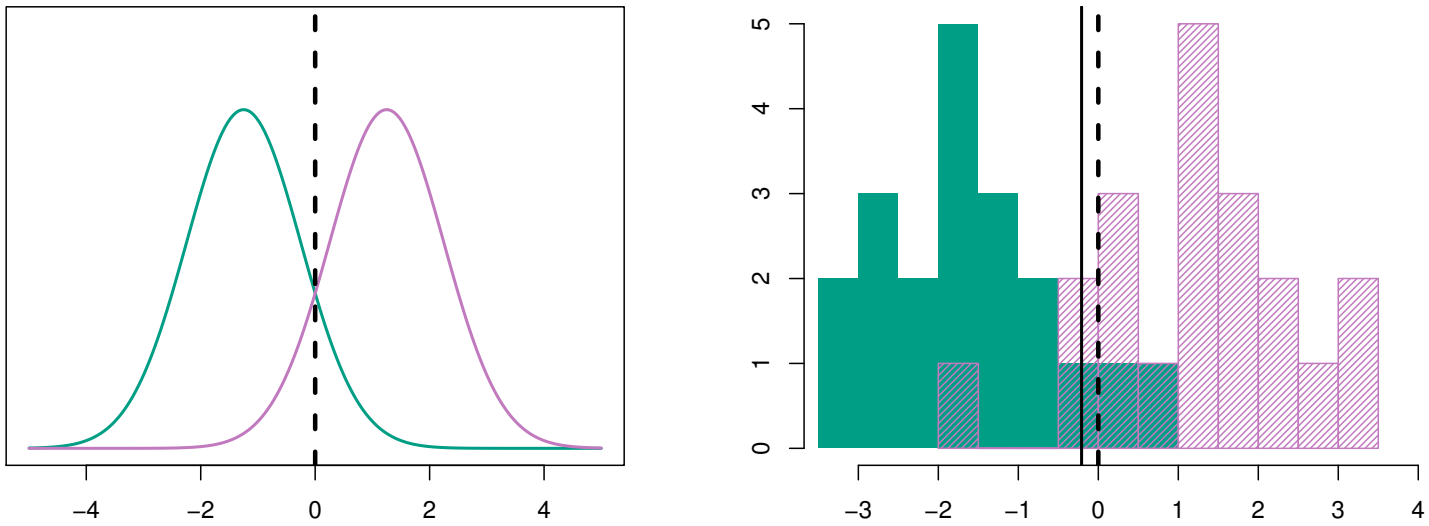


Figure: (Left): Two one-dimensional normal density functions. (Right): 20 observations were simulated from each of the two classes. The dashed black line represents the Bayes decision boundary; the black solid line on the right represents the LDA decision boundary.

Linear Discriminants: general $p > 1$ case

- Suppose there are $p > 1$ predictors and $Y \in \{1, \ldots, m\}$
- **Distributional Assumption**: $X \mid Y = j$ is multivariate Gaussian with mean μ_j and the *same* variance $\text{Cov}(X|Y = j) = \Sigma$:

$$f_j(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_j)^T \Sigma^{-1}(\mathbf{x} - \mu_j)\right)$$

- Then we can obtain the discriminant functions:

$$\delta_j(\mathbf{x}) := \log(\mathbb{P}(Y = j \mid X = \mathbf{x})) = \mathbf{x}^T \Sigma^{-1} \mu_j - \frac{1}{2} \mu_j^T \Sigma^{-1} \mu_j + \log(\pi_j)$$

Linear Discriminant Analysis (LDA) for general p

Suppose that $Y \in \{1, \ldots, m\}$ and $X \in \mathbb{R}^p$. The linear discriminant functions of $Y \mid X = \mathbf{x}$ are:

$$\hat{\delta}_j(\mathbf{x}) = \mathbf{x}^T \widehat{\Sigma}^{-1} \hat{\mu}_j - \frac{1}{2} \widehat{\Sigma}^{-1} \hat{\mu}_j + \log(\hat{\pi}_j), \qquad j = 1, \ldots, m$$

The Bayes decision boundaries are the values of $\mathbf{x}$ for which $\hat{\delta}_j(\mathbf{x}) = \hat{\delta}_\ell(\mathbf{x})$ for $j \neq \ell$, namely where

$$\mathbf{x}^T \widehat{\Sigma}^{-1} \hat{\mu}_j - \frac{1}{2} \hat{\mu}_j^T \widehat{\Sigma}^{-1} \hat{\mu}_j = \mathbf{x}^T \widehat{\Sigma}^{-1} \hat{\mu}_\ell - \frac{1}{2} \hat{\mu}_\ell^T \widehat{\Sigma}^{-1} \hat{\mu}_\ell$$

Summary of LDA

- Overall aim is to identify the Bayes classifier
- To acheive this, we assume that $X \mid Y = j \sim N(\mu_j, \Sigma)$
- We then estimate μ_j and Σ and calculate the discriminant functions $\delta_j(\mathbf{x}) = \log(\mathbb{P}(Y = j \mid X = \mathbf{x}))$
- $\delta_j(\mathbf{x})$ is a linear function of $\mathbf{x}$

Question: In many cases, we don't expect each observation $X \mid Y = j$ to have the same variance Σ. What if we allowed heteroscedacticity?

Quadratic Discriminants: general $p > 1$ case

- Suppose there are $p > 1$ predictors and $Y \in \{1, \ldots, m\}$
- **Distributional Assumption**: $X \mid Y = j$ is multivariate Gaussian with mean μ_j (potentially) *different* variances $\text{Cov}(X|Y = j) = \Sigma_j$:

$$f_j(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_j)^T \Sigma_j^{-1}(\mathbf{x} - \mu_j)\right)$$

- Then we can obtain the discriminant functions (via Bayes):

$$\begin{aligned}\delta_j(\mathbf{x}) &:= \log(\mathbb{P}(Y = j \mid X = \mathbf{x})) \\ &= -\frac{1}{2}\mathbf{x}^T\Sigma_j^{-1}\mathbf{x} + \mathbf{x}^T\Sigma_j^{-1}\mu_j - \frac{1}{2}\mu_j^T\Sigma_j^{-1}\mu_j - \frac{1}{2}\log\left(|\Sigma_j|\right) + \log(\pi_j)\end{aligned}$$

QDA for general p

Suppose that $Y \in \{1, \ldots, m\}$ and $X \in \mathbb{R}^p$. The quadratic discriminant functions of $Y \mid X = \mathbf{x}$ are:

$$\hat{\delta}_j(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\widehat{\Sigma}_j^{-1}\mathbf{x} + \mathbf{x}^T\widehat{\Sigma}_j^{-1}\hat{\mu}_j - \frac{1}{2}\hat{\mu}_j^T\widehat{\Sigma}_j^{-1}\hat{\mu}_j - \frac{1}{2}\log\left(|\widehat{\Sigma}_j|\right) + \log(\hat{\pi}_j)$$

Summary:

- We assume that $X \mid Y = j \sim N(\mu_j, \Sigma_j)$
- We then estimate μ_j and Σ and calculate the discriminant functions $\delta_j(\mathbf{x}) = \log(\mathbb{P}(Y = j \mid X = \mathbf{x}))$
- $\delta_j(\mathbf{x})$ is a quadratic function of $\mathbf{x}$

Linear vs. Quadratic Discriminant Analysis

Why should we ever use one method over another? The answer comes back to the bias / variance tradeoff.

- QDA is much more flexible, which often leads to high variance
  - QDA: $\frac{mp(p+1)}{2}$ parameters
  - LDA: mp parameters
- If the assumption of LDA that each $X \mid Y = j$ has the same covariance structure is wrong, then LDA will much higher bias than QDA.
- **Conclusion:** It again depends on the data and for the user to check assumptions.

Next Up

- Computational Methods for Bayesian Analysis
- Imputation
- Bayesian Hypothesis Testing