[POLS 8500] Regularization, Dimensionality Reduction, Classification Problems

L. Jason Anastasopoulos ljanastas@uga.edu

February 2, 2017

Methods to decrease the number of features without decreasing model accuracy

- **Subset Selection**- find $p \subset P$ that best fit Y.

Methods to decrease the number of features without decreasing model accuracy

- **Subset Selection**- find $p \subset P$ that best fit Y.
- **Shrinkage/Regularization** - Fit model to all p predictors, shrink values of some $\theta \to 0$.

Methods to decrease the number of features without decreasing model accuracy

- **Subset Selection**- find $p \subset P$ that best fit Y.
- **Shrinkage/Regularization** - Fit model to all p predictors, shrink values of some $\theta \rightarrow 0$.
- **Dimensionality Reduction** - Project p onto $\mathcal{M}$ dimensional subspace s.t. $\mathcal{M} < p$. Use $\mathcal{M}$ as predictors.

Methods of subset selection

- Best subset.

Methods of subset selection

- Best subset.
- Forward stepwise.

Methods of subset selection

- Best subset.
- Forward stepwise.
- Backward stepwise.

Methods of subset selection

- All are essentially a means of reducing the number of variables that you include in a model.
- An alternative to this is the inclusion of all of the predictors that you'd like, but have a model that *constrains* or *regularizes* the coefficient values.

Regularization and shrinkage

- *Regularization* methods effectively downweight coefficients toward zero.
- This has the effect of improving fit and reducing the variance of estimates.
- Two techniques we will be discussing are *ridge regression* and the *lasso*.

Ridge regression

Recall that for linear regression we seek to estimate parameter values that minimize the cost function:

$$J(\theta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2$$

Ridge regression

- In ridge regression we seek to do the same thing except now we add $\lambda \sum_{j=1}^{p} \theta_j$

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \theta_j^2$$

Ridge regression

- In ridge regression we seek to do the same thing except now we add $\lambda \sum_{j=1}^{p} \theta_j$

- So that now we're minimizing

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \theta_j^2$$

Ridge regression

- $\lambda > 0$ and is referred to as a *tuning parameter* and is determined independently.

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\theta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\theta_j^2$$

Ridge regression

- $\lambda > 0$ and is referred to as a *tuning parameter* and is determined independently.

- $\lambda \sum_{j=1}^{p} \theta_j$ is called the *shrinkage penalty*

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \theta_j^2$$

Shrinkage penalty

$$\lambda \sum_{j=1}^{p} \theta_j$$

- When $\theta_1, \cdots, \theta_p$ are close to zero the shrinkage penalty shrinks them toward zero.
- When $\lambda = 0$ we estimate OLS coefficients.
- As $\lambda \to \infty$ all coefficients shrink to zero.

Shrinkage penalty

$$\lambda \sum_{j=1}^{p} \theta_j$$

- Shirinkage does not affect the intercept θ_0, only the coefficients with variables attached to them.
- It essentially downweights the impact that some variables have.

Shrinkage penalty

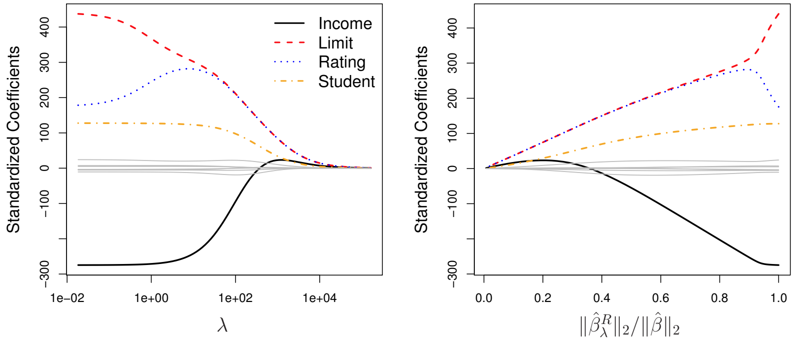


Figure 1: optional caption text

- Shrinkage does not affect the intercept θ_0, only the coefficients with variables attached to them.

Ridge regression - issues

$$x_i = \frac{x_i}{s}$$

- Ridge regression is sensitive to the scaling of the predictiors.
- May get different outcomes if your predictor is **X** vs **1000X**.
- Predictors should be *standardized* to avoid these issues.

Ridge regression and the bias-variance tradeoff

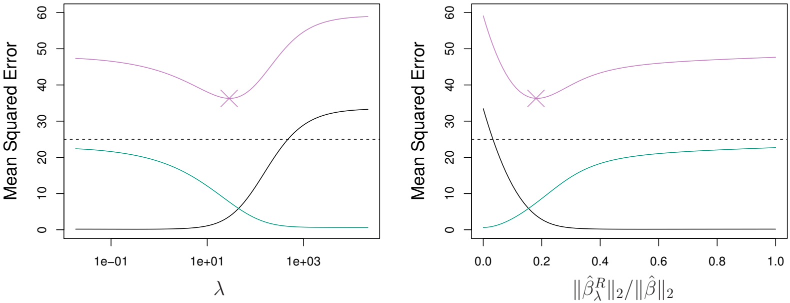


Figure 2: Squared bias (black), variance (green), and test MSE (purple) for ridge regression as a function of λ

- Ridge regression can lead to significant reductions in test mean squared error over OLS with only small increases in bias.
- Ridge regression also less computationally intensive than some

The Lasso

- Ridge regression will shrink coefficients toward zero.
- Will not set coefficients to zero unless $\lambda = \infty$.
- This can be a problem for model interpretation when p is large.
- All predictors will be retained.

The Lasso

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\theta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\theta_j|$$

- Lasso solves this problem with a very simple tweak.
- The L_2 penalty of the ridge regression is replaced with the L_1 penalty of the Lasso.
- For λ sufficiently large some $\theta_j = 0$.
- Produces *sparse* models.

Why does the Lasso produce estimates that $= 0$?

- It is not entirely clear why the Lasso would produce coefficient estimates that equal zero while ridge regression does not.
- To understand why this is the case it is necessary to frame Lasso and Ridge regression as *contrained optimization* problems (remember Langrange multipliers!)

Constrained optimization

Maximize $f(x, y)$

Subject to $g(x, y) = c$

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

- By calculating the gradient of the Lagranginan $\nabla\mathcal{L}(x, y, \lambda) = 0$, solving for λ, x and y in the system of equations and assessing $f(x, y)$ at the critical points will give you the minima and maxima subject to the constraints.

Constrained optimization: example

Maximize $f(x, y) = x + y$

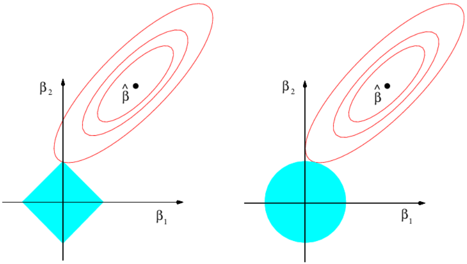
Subject to $g(x, y) = x^2 + y^2 = 1$

Lasso and ridge regression are constrained optimization problems

$$\text{Lasso:}\quad \arg\min_{\theta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 \quad \text{subject to:} \sum_{j=1}^{p} |\theta_j| \leq s$$

$$\text{Ridge:}\quad \arg\min_{\theta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 \quad \text{subject to:} \sum_{j=1}^{p} (\theta_j)^2 \leq s$$

Lasso and ridge regression are constrained optimization problems



Two feature model. Contour plot of error and constraint functions for lasso (left) and ridge regression (right). Notice that the error functions and constraint for the lasso meet at $\beta_1 = 0$.

Comparing Lasso and Ridge regression

$$Y = \theta_0 + \sum_i \theta_i x_i$$

- Lasso is better for interpretability, but the predictive model may not be better if many of the predictors truly do not equal zero.
- Take the model above. If, for $i = 1, \cdots, 20$ for example and $\theta_i > 0$, then ridge regression will outperform the Lasso.

Bayesian interpretations

$$p(\theta|X, Y) \propto f(Y|X, \theta)p(\theta)$$

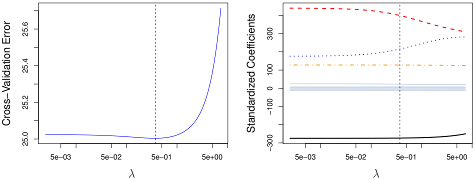
- Ridge regression and the Lasso can also be thought of as Bayesian models.

- If we assume $\theta \sim N(\mu = 0, \sigma^2 = \lambda)$ and $p(\theta) = \prod_p \Phi(\mu = 0, \sigma^2 = \lambda)$, the posterior mode for θ is the ridge regression solution.
- If we assume $\theta \sim \Lambda(\mu = 0, b = \lambda)$ and $p(\theta) = \prod_p \Lambda(\mu = 0, b = \lambda)$, the posterior mode for θ is the lasso solution.

Selecting λ

- Choose a range of values for $\lambda \in [a, b]$
- For each $\lambda_i \in [a, b]$ calculate the cross-validated error CV_i.
- Select λ_i such that min CV_i, the cross-validated error is minimized.

Selecting λ



Left: Cross validated errors from applying ridge regression to the **Credit** data set with several values of λ. Right: standardized coefficient values as a function of λ.

Dimensionality Reduction Methods

- All of the models discussed used the original predictors in some form.
- Dimensionality reduction methods transform the predictors into variable clusters and then use these transformed variables to fit a model.

Dimensionality Reduction Methods

Consider a linear combination $Z_1, \cdots, Z_M$ of the features $X_1, \cdots, X_1$ such that $M < p$ where:

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

For some constants $\phi_1, \cdots, \phi_M; m \in [1, M]$. We can then fit the linear regression model:

$$y_i = \Theta_0 + \sum_{m=1}^{M} \Theta_m z_{im}$$

Dimensionality Reduction Methods

The model

$$y_i = \Theta_0 + \sum_{m=1}^{M} \Theta_m z_{im}$$

now has $M + 1 < p + 1$ predictors and, if chosen well, can result in a better fit through estimating fewer parameters than the original regression model.

Dimensionality Reduction Methods

To be clear take a simple linear regression model with three features:

$$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \epsilon$$

Define $z_1 = \phi_1 X_1 + \phi_3 X_3$ and $z_2 = \phi_2 X_2$. We can now estimate the reduced model:

$$\begin{aligned} Y &= \Theta_0 + \Theta_1 z_1 + \Theta_2 z_2 + \epsilon \\ &= \Theta_0 + \Theta_1(\phi_1 X_1 + \phi_3 X_3) + \Theta_2(\phi_2 X_2) + \epsilon \end{aligned}$$

Dimensionality Reduction Methods

- Again the key here is that we are estimating a model with fewer predictors, thus reducing the *dimensionality* of the model.
- This is especially useful in problems where p is large relative to n. Variance will be significantly reduced in this case and this is not uncommon in machine learning problems (ie text analysis)

All dimensionality reduction methods involves two steps

1. Transformed predictors $Z_1, \cdots, Z_M$ are first obtained.

All dimensionality reduction methods involves two steps

1. Transformed predictors $Z_1, \cdots, Z_M$ are first obtained.

2. A model is fit using the M predictors.

All dimensionality reduction methods involves two steps

1. Transformed predictors $Z_1, \cdots, Z_M$ are first obtained.

2. A model is fit using the M predictors.

- There are several methods for accomplishing this but we will focus on principal components analysis.

Principal Components Analysis (PCA)

$$f : \mathcal{X} \rightarrow \mathcal{F}$$

$$\mathcal{X} \in \mathbb{R}^{nxp}, \mathcal{F} \in \mathbb{R}^{nxm}; p << m$$

- PCA is often discussed in the context of *unsupervised learning* and we'll discuss it in that context later on in the semester.
- It's a popular means of transforming a high dimensional feature space $\mathcal{X}$ into a very low-dimensional space $\mathcal{F}$

Principal Components Analysis (PCA)

- **First principal component** is the dimension along which the data vary the most and would be the most useful for a regression approach.

```
# Predicting political party with votes
library(mlbench)
data(HouseVotes84)
head(HouseVotes84)
```

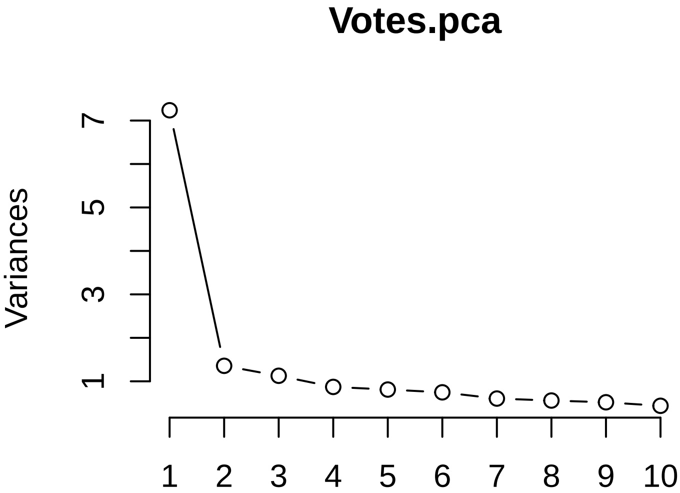
```
##        Class   V1 V2 V3   V4   V5 V6 V7 V8 V9 V10  V11  V12 V13 V14
## 1 republican    n  y  n    y    y  y  n  n  n   y <NA>    y   y   y
## 2 republican    n  y  n    y    y  y  n  n  n   n    n    y   y   y
## 3   democrat <NA>  y  y <NA>    y  y  n  n  n   n    y    n   y   y
## 4   democrat    n  y  y    n <NA>  y  n  n  n   n    y    n   y   n
## 5   democrat    y  y  y    n    y  y  n  n  n   n    y <NA>   y   y
## 6   democrat    n  y  y    n    y  y  n  n  n   n    n    n   y   y
##    V16
## 1    y
## 2 <NA>
## 3    n
## 4    y
## 5    y
```

Predicting political party from votes, 1984

```
##
## Call:
## lm(formula = Party ~ ., data = data.frame(Votes))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82054 -0.04439  0.01879  0.08784  0.70224
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.7671523  0.1423941   5.388 1.20e-07 ***
## V1          -0.0264965  0.0213872  -1.239 0.216080
## V2          -0.0282866  0.0200216  -1.413 0.158459
## V3          -0.1988778  0.0296838  -6.700 6.76e-11 ***
## V4           0.6314606  0.0322082  19.606  < 2e-16 ***
## V5           0.0768241  0.0379082   2.027 0.043339 *
## V6          -0.0481934  0.0272663  -1.768 0.077872 .
## V7           0.0642615  0.0288318   2.229 0.026355 *
## V8           0.0559900  0.0352681   1.588 0.113144
## V9          -0.0855327  0.0314344  -2.721 0.006780 **
## V10          0.0478126  0.0187176   2.554 0.010990 *
## V11         -0.1240756  0.0199903  -6.207 1.30e-09 ***
```

Predicting political party from votes, 1984

- Can the votes be explained with a single dimension?



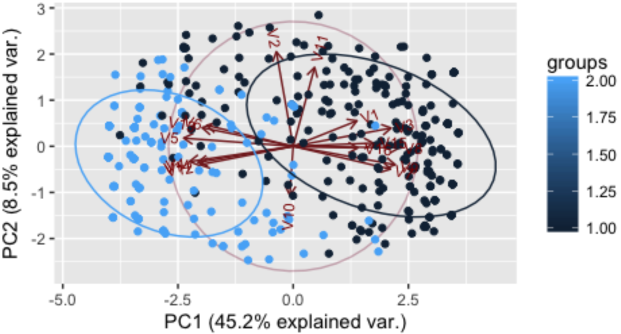
Predicting political party from votes, 1984

- Can the votes be explained with a single dimension?

```
summary(Votes.pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC
## Standard deviation     2.6901 1.16470 1.06151 0.93320 0.90006 0.8638
## Proportion of Variance 0.4523 0.08478 0.07043 0.05443 0.05063 0.0466
## Cumulative Proportion  0.4523 0.53706 0.60749 0.66192 0.71255 0.7591
##                            PC7     PC8     PC9    PC10    PC11    PC
## Standard deviation     0.77631 0.74664 0.71935 0.66043 0.64191 0.576
## Proportion of Variance 0.03767 0.03484 0.03234 0.02726 0.02575 0.020
## Cumulative Proportion  0.79686 0.83170 0.86404 0.89130 0.91705 0.937
##                           PC13    PC14    PC15    PC16
## Standard deviation     0.56507 0.52220 0.48542 0.40914
## Proportion of Variance 0.01996 0.01704 0.01473 0.01046
## Cumulative Proportion  0.95777 0.97481 0.98954 1.00000
```

Predicting political party from votes, 1984



Predicting political party from votes, 1984

- Took 16 dimensions, reduced to 1 or 2 that still explain about 50% of the variance.
- Can use these dimensions in regression for comparison.
- Let's just use dimensions one and two

Predicting political party from votes, 1984

$$Party = \Theta_0 + \Theta\pi_1 + \Theta_2\pi_2$$

- Took 16 dimensions, reduced to 1 or 2 that still explain about 50% of the variance.
- Can use these dimensions in regression for comparison.
- Let's just use dimensions one and two.

Predicting political party from votes, 1984

$$Party = \Theta_0 + \Theta\pi_1 + \Theta_2\pi_2$$

```
pi1<-Votes.pca$x[,1]
pi2<-Votes.pca$x[,2]
summary(lm(Party~pi1 + pi2))
```

```
##
## Call:
## lm(formula = Party ~ pi1 + pi2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9021 -0.1181  0.0211  0.1560  0.9177
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.386207   0.012918 107.312   <2e-16 ***
## pi1         -0.144037   0.004807 -29.961   <2e-16 ***
## pi2         -0.105903   0.011104  -9.538   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problems with PCA

- Very sensitive to scaling
- Is a good idea to standardize the predictors.

For next time

- Machine learning methods for classification: logistic regression, linear discriminant analysis, k-means clustering, naive bayes.
- Introduction to text analysis.

Classification

- Linear regression assumes that the outcome Y is quantitative.
- But in most machine learning situations the outcome variable that we're interested in is a class or category of something.
- This is especially true in text analysis where we want to classify documents etc.
- In machine learning, algorithms that conduct classification tasks are known as *classifiers* and we will discuss three of the most popular classifiers: (1) logistic regression, (2) linear discriminant analysis and (3) K-nearest neighbors.

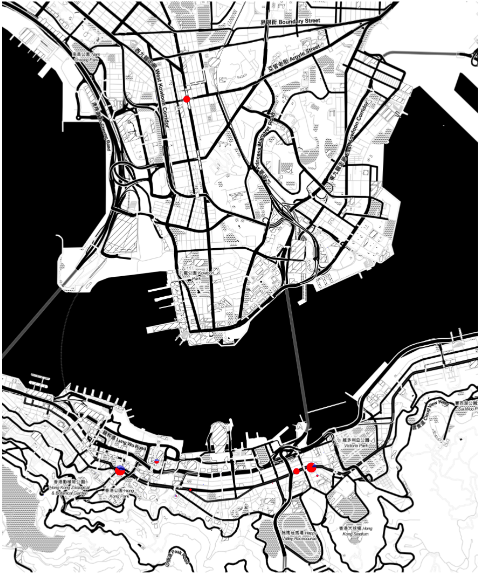
Classification problems in the real world

- Likelihood of someone having a disease given personal charachteristics, syptoms etc.
- Recommender systems: what will *Netflix* recommend given things that you've rated and your personal charachteristics?
- What kind of alleles (DNA polymorphisms) are associated with the likelihood of getting a disease? (ie BRCA and breast cancer?)

Classification problems in social science

- Can we predict political party in the House and Senate based on voting behavior.
- Conversely, can we predict how someone will vote on a bill given past voting behavior, political party and other charachteristics.
- What is the race of an individual given their facial photo?
- Does a document contain violence according to some standard (WHO etc?)
- How wide ranging is the scope of an executive order?

Do Tweets contain either descriptions or exhortations to violence?



Why not use linear regression for classification?

Tweet Codings: 1 = Description of violence, 2 = Exhortation to violence, 3 = No discussion of violence.

- If we use linear regression in this context, the model will infer that there is some kind of meaningful ordering attached to the codings.
- That 3 is *more than* 2 and 1.
- When in reality all we are doing is using these as placeholders.

Why not use linear regression for classification?

- We could collapse the categories to 1 = Description OR exhortation to violence, and 0 = No discussion of violence.

$$P(Violence|W) = \theta_0 + \sum_w \theta_w x_w + \epsilon$$

- In this case, regression makes some sense but because this kind of outcome is not continuous we will get $P(Violence|W) < 0$ and $P(Violence|W) > 1$ which does not make sense.
- In the case of a binary predictor, we want to use *logistic regression* which models this situation with greater interpretability and more accurately reflects the way the data is structured.

Logistic regression

$$P(Violence|W) > 0.5\text{: Tweet is violent}$$
$$P(Violence|W) \leq 0.5\text{: Tweet is not violent}$$

- Consider the situation in which we're interested in figuring out whether a Tweet contains some linguistic information which suggests that violence is being referenced or not.
- In this case we would want to build a model which uses the words in a Tweet and gives us a probability of containing some violent content.
- We then set some threshold to