Penalized regression methods

Patrick Breheny

February 20
Minimizing the residual sum of squares

- Linear regression revolves around minimizing the residual sum of squares:

\[
\text{RSS} = \sum_{i} (y_i - x_i^T \beta)^2
\]

- The value that minimizes this function, \( \hat{\beta} \), is often a good estimator of \( \beta \)

- There are many situations, however, where it isn’t a very good estimator – when the number of parameters is large, or when columns of \( X \) are highly correlated – and instead produces unstable, highly variable estimates
Penalized regression

One way of dealing with this problem is to introduce a penalty: instead of minimizing RSS, we minimize

$$RSS(\beta) + P(\beta),$$

where $P$ is a function that penalizes what one would consider less realistic values of the unknown parameters (more on this later).

Penalized methods always come with at least one *regularization parameter*, here $\lambda$, that controls the tradeoff between likelihood and penalty.
Correlated predictors

As an example of how this works, suppose that the expected value of an outcome $Y$ depends on two other factors $x_1$ and $x_2$:

$$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

However, suppose that $x_1$ and $x_2$ are highly correlated:

```r
> x1 <- rnorm(20); x2 <- rnorm(20, mean=x1, sd=.01)
> cor(x1, x2)
[1] 0.9999423
> y <- rnorm(20, mean=3+x1+x2)
> coef(lm(y~x1+x2))
(Intercept)    x1       x2
  2.582064  39.971344 -38.040040
```
Ridge regression

Why do we obtain such an awful estimate?

Because $x_1$ and $x_2$ are so correlated, $\text{RSS}(40, -38) = 21.7$ (our estimate) is very close to $\text{RSS}(1, 1) = 22.6$ (the truth), as is any solution for which $\beta_1 + \beta_2 = 2$.

A very effective way of dealing with this problem is through penalization: instead of minimizing RSS, we minimize

$$\text{RSS}(\beta) + \lambda \sum_{j=1}^{p} \beta_j^2$$

This approach is known as ridge regression.
Ridge regression (cont’d)

- **Why does this help?**
  
  Well, \( \text{RSS}(1, 1) \approx \text{RSS}(40, -38) \), but \( 1^2 + 1^2 = 2 \) is much less than \( 40^2 + (-38)^2 = 3044 \).

- **The solution \((1, 1)\) will incur a much smaller penalty and be greatly favored over \((40, -38)\):**

```r
> lm.ridge(y~x1+x2,lambda=1)
x1     x2
2.6214998 0.9906773 0.8973912
```
**Theorem:** The solution to the ridge regression problem is given by

$$\hat{\beta} = (X^TX + \lambda I)^{-1}X^Ty$$

Note the similarity to the ordinary least squares solution, but with the addition of a “ridge” down the diagonal

**Corollary:** As $\lambda \to 0$, $\hat{\beta}^{\text{ridge}} \to \hat{\beta}^{\text{OLS}}$

**Corollary:** As $\lambda \to \infty$, $\hat{\beta}^{\text{ridge}} \to 0$
Corollary: In the special case of an orthonormal design matrix,

\[ \hat{\beta}_J^{\text{ridge}} = \frac{\hat{\beta}_J^{\text{OLS}}}{1 + \lambda} \]

- This illustrates the essential feature of ridge regression: \textit{shrinkage}
- Applying the ridge regression penalty has the effect of shrinking the estimates toward zero – introducing bias but reducing the variance of the estimate
Recall that ordinary least squares estimates do not always exist; if $\mathbf{X}$ is not full rank, $\mathbf{X}^T \mathbf{X}$ is not invertible and there is no unique solution for $\hat{\beta}$.

This problem does not occur with ridge regression, however.

**Theorem:** For any design matrix $\mathbf{X}$, the quantity $\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}$ is always invertible; thus, there is always a unique solution $\hat{\beta}$. 
Existence Theorem: For any value of $\beta$, there always exists a $\lambda$ such that the MSE of $\hat{\beta}_\text{ridge}_\lambda$ is less than the MSE of $\hat{\beta}_\text{OLS}$.

This is a rather surprising result with somewhat radical implications: even if the model we fit is exactly correct and follows the exact distribution we specify, we can always obtain a better estimator by shrinking towards zero.
Suppose we have a number of potentially important predictors, but that we suspect some are unrelated to the outcome.

Direct minimization of RSS for the full model is not particularly desirable, as it will produce a $\hat{\beta}$ with a large variance.

Probably the most common solution to this problem is to choose a subset of important explanatory variables based on either the significance of the terms in the model or some sort of criterion such as AIC.
Limitations of subset selection

- One problem with this approach is that the number of “all possible subsets” grows exponentially with $p$ and is not computationally feasible for $p$ much larger than about 40 or 50.
- Another problem is the fact that subset selection is discontinuous, in the sense that an infinitesimally small change in the data can result in completely different estimates.
- As a result, subset selection is often unstable and highly variable.
A different way of dealing with this problem is to use penalized regression. However, the ridge regression penalty ($\sum \beta_j^2$), although it helps with obtaining less variable estimates, has two big shortcomings in this setting:

- Heavy bias toward zero for large regression coefficients
- Interpretability: unimportant coefficients may be shrunken towards zero, but they’re still in the model
Consider instead a different penalized regression approach, that of minimizing

\[ \frac{1}{2} \text{RSS}(\beta) + \lambda \sum |\beta_j|, \]

the only difference from ridge regression being that absolute values, instead of squares, are used in the penalty function.

The change to the penalty function is subtle, but has a dramatic impact on the resulting estimator.
Like ridge regression, penalizing the absolute values of the coefficients introduces shrinkage towards zero.

However, unlike ridge regression, some of the coefficients are shrunk all the way to zero; such solutions, with multiple values that are identically zero, are said to be *sparse*.

The penalty thereby performs a sort of continuous variable selection.

The resulting estimator was thus named the *lasso*, for “Least Absolute Shrinkage and Selection Operator.”
Orthonormal Solutions

Because the lasso penalty has the absolute value operation in it, the objective function is not differentiable and as a result, lacks a closed form in general.

However, in the special case of an orthonormal design matrix, it is possible to obtain closed form solutions for the lasso: 

\[
\hat{\beta}_{\text{lasso}}^j = S(\hat{\beta}_{\text{OLS}}^j, \lambda),
\]

where \( S \), the soft-thresholding operator, is defined as:

\[
S(z, \lambda) = \begin{cases} 
  z - \lambda & \text{if } z > \lambda \\
  0 & \text{if } |z| \leq \lambda \\
  z + \lambda & \text{if } z < -\lambda
\end{cases}
\]
A geometric illustration of why lasso results in sparsity, but ridge does not, is given by the constraint interpretation of their penalties:

**FIGURE 3.11.** Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.
Another way of seeing how the lasso produces sparsity is to view it from a Bayesian perspective, where the lasso penalty arises from a double exponential prior:

Note that the lasso prior is “pointy” at 0, so there is a chance that the posterior mode will be identically zero.
The function on the previous slide is referred to as “soft” thresholding to distinguish it from *hard thresholding*:

\[ H(z, \lambda) = \begin{cases} 
  z & \text{if } |z| > \lambda \\
  0 & \text{if } |z| \leq \lambda
\end{cases} \]

In the orthonormal case, best subset selection is equivalent to hard thresholding.

Note that soft thresholding is continuous, while hard thresholding is not.
Thus, in the orthonormal case, each of the methods we have discussed are simple functions of the least squares solutions:

\[
\begin{align*}
\text{Subset selection:} & \quad \hat{\beta}_j = H(\hat{\beta}^{OLS}_j, \lambda) \\
\text{Ridge:} & \quad \hat{\beta}_j = \frac{\hat{\beta}^{OLS}_j}{1 + \lambda} \\
\text{Lasso:} & \quad \hat{\beta}_j = S(\hat{\beta}^{OLS}_j, \lambda)
\end{align*}
\]
Ridge vs. lasso coefficient paths

For non-orthogonal problems, the result is more complex (we will look at this data set later, in lab)
Earlier, we mentioned that penalized regression revolves around an assumption that coefficient values around zero are more believable than those far away from zero. Some care is needed, however, in the application of this idea. First of all, it does not make sense to apply this idea to the intercept (unless you happened to have some reason to think that the mean of $y$ should be zero). Hence, the intercept is not included in the penalty; if it were, the estimates would not be invariant to changes of location.
A separate consideration is how to make “far from zero” mean the same thing for all the variables

For example, suppose $x_1$ varied from 0 to 1, while $x_2$ varied from 0 to 1 million; clearly, a one-unit change in $x$ does not mean the same for both of these variables

Thus, the explanatory variables are usually standardized prior to model fitting to have mean zero and standard deviation 1; i.e.,

$$\bar{x}_j = 0$$

$$x_j^T x_j = n$$

for all $j$
Summary

- In practice, both exclusion of the intercept and standardization are accomplished internally by whatever software you are using to fit the model.

- To clarify: by “internally”, I mean that the software rescales the data, fits the model to the standardized data, then transforms back in order to provide estimates on the original scale of the variables.

- The key thing to note is that after standardization and exclusion of the intercept from the penalization, we have results that are location-scale invariant.
Obviously, penalized regression methods depend heavily on the regularization coefficient $\lambda$; how should we select it?

There are various methods, but the most widely used is based on how well predictions based on $\hat{\beta}_\lambda$ do at predicting actual instances of $Y$.

Now, it would not be fair to use the data twice – once to fit the model and then again to estimate the prediction accuracy – as this would reward overfitting.
One idea is to split the data set into two fractions, then use one portion to fit $\hat{\beta}$ and the other to evaluate how well $X\hat{\beta}$ predicted the observations in the second portion.

The problem with this solution is that we rarely have so much data that we can freely part with half of it solely for the purpose of choosing $\lambda$.

To finesse this problem, *cross-validation* splits the data into $K$ folds, fits the data on $K - 1$ of the folds, and evaluates risk on the fold that was left out.
This process is repeated for each of the folds, and the risk averaged across all of these results:

Common choices for $K$ are 5, 10, and $n$ (also known as leave-one-out cross-validation)
Example: Nonparametric regression

One application of the idea of penalized regression is in the estimation of regression curves, where excessively “wiggly” functions are penalized:

\[ P_\lambda(f) = \lambda \int \{f''(x)\}^2 dx \]

\( \lambda \) too big

\( \lambda \) just right

\( \lambda \) too small
Effect of CAFO exposure on risk of childhood asthma
Another important application of penalized regression methods is when the number of potential predictors is very large. This is a problem that comes up increasingly often as it becomes easier to generate data in an automated fashion and cheaper to store and transfer it. These sorts of applications are particularly common in modern biological and medical studies, where much research centers around trying to predict an outcome (blood pressure, disease status, response to treatment) based on molecular features such as genes, DNA patterns, metabolites, proteomic signatures, methylation patterns, ...
The most serious flaw of the lasso is its bias toward zero for large regression coefficients (its bias is not as large as ridge regression, but can still be improved upon). The MCP is an example of a penalty that minimizes this bias.
Another extension I have worked on quite a bit is when covariates are organized into groups.

For example, a categorical variable may be represented by a group of indicator functions, a continuous variable may be represented by a group of basis functions, or variables might be grouped for scientific reasons (e.g., genetic markers can be grouped by the genes to which they belong).

This can be reflected in a hierarchical penalty:

\[ P_\lambda(\beta) = \sum_{j=1}^{p} P_{O,\lambda_1} \left\{ \sum_{k=1}^{K_j} P_{I,\lambda_2}(\beta_{jk}) \right\} , \]

where \( P_O \) denotes a group-level penalty and \( P_I \) an individual-level penalty.
Prostate cancer study

- An an example, consider the data from a 1989 study examining the relationship prostate-specific antigen (PSA) and a number of clinical measures in a sample of 97 men who were about to receive a radical prostatectomy.
- PSA is typically elevated in patients with prostate cancer, and serves a biomarker for the early detection of the cancer.
- The explanatory variables:
  - lcavol: Log cancer volume
  - lweight: Log prostate weight
  - age
  - lbph: Log benign prostatic hyperplasia
  - svi: Seminal vesicle invasion
  - lcp: Log capsular penetration
  - gleason: Gleason score
  - pgg45: % Gleason score 4 or 5
We’ll now look at applying ridge, lasso, and MCP to this data set using various R packages; the code to follow along is available at http://web.as.uky.edu/statistics/users/pbreheny/603.R
To see how these methods can be applied to large data sets, let’s also use the lasso/MCP to analyze a study of breast cancer metastasis published by Van ’t Veer et al. in 2002.

In the study, biological samples were obtained from the tumors of women with breast cancer.

These samples were applied to a small chip called a microarray, which can measure the expression of tens of thousands of genes (i.e. how much of the gene is being produced by the cells in the sample).

The patients were then followed to see how long it took for the cancer to metastasize (spread).

Clinically, the goal is to identify patients with poor prognosis in order to administer more aggressive follow-up treatment.
Van ’t Veer study: Some results

- In the study, there were 98 subjects and 24,188 genes.
- Of those genes, the lasso selected 13 as being useful for predicting metastasis (i.e., 13 of the regression coefficients were nonzero, the other 24,175 were equal to zero).
- Using this model, the lasso was able to correctly predict whether cancer would recur in the next 5 years for 67.3% of the patients.
- MCP achieves a similar prediction performance using only 5 genes.