Week 6:
Variable Selection and Regularization

CMPS 320: Machine Learning

Outline

1. Introduction

2. Subset Selection

3. Shrinkage Methods

Introduction

- In the regression setting, the standard linear model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon \tag{1}$$

  is commonly used to describe the relationship between a response Y and a set of variables $X_1, X_2, \cdots, Xp$.
- The coefficients $\beta_1, \beta_2, \cdots, \beta_p$ are estimated using the least square.
- In this section, we consider other fitting procedures.
- We will show that alternative fitting procedures can yield better
  - prediction accuracy and
  - model interpretability.

Prediction accuracy

- If the true relationship between the response and the predictors is approximately linear, the least squares estimates will have low bias.
- If $n >> p$, that is, if the number of observations, n, is much larger than the number of variables, p then the least squares estimates tend to also have low variance, and hence will perform well on test observations.
  - However, if n is not much larger than p, then there can be a lot of variability in the least squares fit, resulting in overfitting and consequently poor predictions on future observations not used in model training.
- If $p > n$, then there is no longer a unique least squares coefficient estimate: the variance is infinite so the method cannot be used at all.
  - By constraining or shrinking the estimated coefficients, we can often substantially reduce the variance at the cost of a negligible increase in bias.

Model Interpretability

- Some or many of the variables used in a multiple regression model are in fact not associated with the response.
- Including such irrelevant variables leads to unnecessary complexity in the resulting model.
    - By removing these variables – that is, by setting the corresponding coefficient estimates to zero, we can obtain a model that is more easily to interpret.
- We will discuss some approaches for automatically performing feature selection or variable selection–that is, excluding irrelevant variables from a multiple regression model.

Subset Selection

- This approach involves identifying a subset of the p predictors that we believe to be related to the response.
- We then fit a model using least squares on the reduced set of variables.
- We will discuss two techniques:
  - Best Subset Selection
  - Stepwise Selection

Best Subset Selection

- To perform best subset selection, we fit a separate least squares regression best subset for each possible combination of the p predictors.
- In particular, we fit all p models selection that contain exactly one predictor, all $\binom{p}{2} = p(p-1)/2$ models that contain exactly two predictors, and so forth.
- We then look at all of the resulting models, with the goal of identifying the one that is best.

Best Subset Selection–Model Overload

- Number of possible models on a set of p predictors:

$$\sum_{k=1}^{p} \binom{p}{k} = 2^p$$

  - On 10 predictors: 1,024 models
  - On 20 predictors: 1,048,576 models
- The best subset selection becomes computationally infeasible for values of p greater than around 40, even with extremely fast modern computers.
- Question: what happens to our estimated coefficients as we fit more and more models?
- Answer: the larger the search space, the larger the variance.
  - We're overfitting!

Forward Stepwise Selection

- Forward stepwise selection is a computationally efficient alternative to best subset selection.
- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- At each step the variable that gives the greatest additional improvement to the fit is added to the model.

Forward Stepwise Selection–Model Overload

- Unlike best subset selection, which involved fitting 2^p models, forward stepwise selection involves fitting one null model, along with $p-k$ models in the kth iteration, for $k = 0, \cdots, p-1$.
- This amounts to a total of $1 + \sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models.
- When $p = 20$, best subset selection requires fitting 1,048,576 models, whereas forward stepwise selection requires fitting only 211 models.
- Question: what potential problems do you see?
- There's a risk we might prune an important predictor too early.
  - While this method usually does well in practice, it is not guaranteed to give the optimal solution.

Backward Stepwise Selection

- Unlike forward stepwise selection, backward stepwise selection begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

Backward Stepwise Selection–Model Overload

- The backward selection approach searches through only $1 + p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection.
- Backward selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit).
- Question: what potential problems do you see?
- Answer: if we have more predictors than we have observations, this method won't work (why?)

Choosing the optimal model

- Best subset selection, forward selection, and backward selection result in the creation of a set of models, each of which contains a subset of the p predictors.
- Measures of training error (RSS and R^2) aren't good predictors of test error (this is what we care about).
- Therefore, RSS and R^2 are not suitable for selecting the best model among a collection of models with different numbers of predictors.

Adjusted R^2

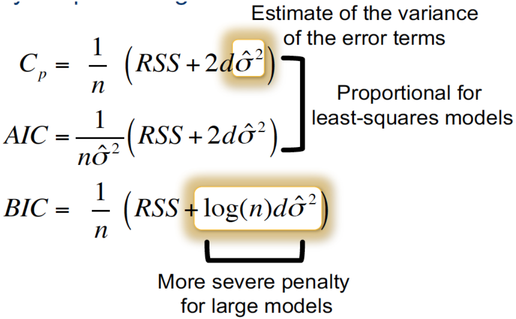
- Intuition: once all of the useful variables have been included in the model, adding additional junk variables will lead to only a small decrease in RSS.

$$R^2 = 1 - \frac{RSS}{TSS} \rightarrow R^2_{Adj} = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

- Adjusted R^2 pays a penalty for unnecessary variables in the model by dividing RSS by $(n - d - 1)$ in the numerator.

C_p, Akaike information criterion (AIC) and Bayesian information criterion (BIC)

- Some other ways of penalizing RSS



Estimate of the variance of the error terms

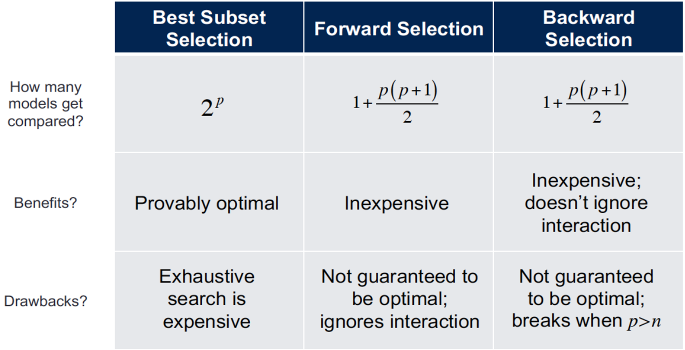
$$C_p = \frac{1}{n}\left(RSS + 2d\boxed{\hat{\sigma}^2}\right)$$

$$AIC = \frac{1}{n\hat{\sigma}^2}\left(RSS + 2d\hat{\sigma}^2\right)$$

Proportional for least-squares models

$$BIC = \frac{1}{n}\left(RSS + \underbrace{\boxed{\log(n)d\hat{\sigma}^2}}_{\text{More severe penalty for large models}}\right)$$

Comparing methods



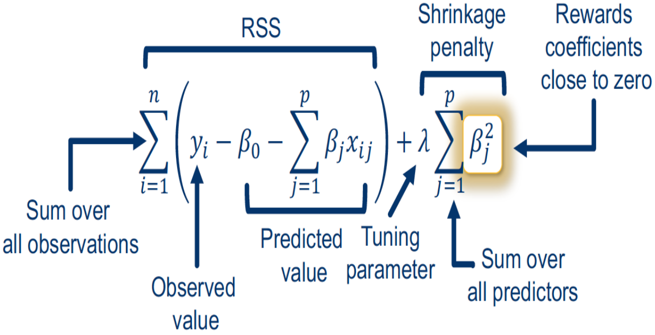
	Best Subset Selection	Forward Selection	Backward Selection
How many models get compared?	2^p	$1+\frac{p(p+1)}{2}$	$1+\frac{p(p+1)}{2}$
Benefits?	Provably optimal	Inexpensive	Inexpensive; doesn't ignore interaction
Drawbacks?	Exhaustive search is expensive	Not guaranteed to be optimal; ignores interaction	Not guaranteed to be optimal; breaks when $p>n$

Shrinkage Methods

- The subset selection methods described in the previous section involve using least squares to fit a linear model that contains a subset of the predictors.
- An alternative approach is to fit a model containing all p predictors using a technique that shrinks the coefficient estimates towards zero.
- It turns out that shrinking the coefficient estimates can significantly reduce their variance.
- The two best-known techniques for shrinking the regression coefficients towards zero are:
    - ridge regression and
    - the lasso

Ridge regression

- Ridge regression is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity.
- Big idea: minimize RSS plus an additional penalty that rewards small (sum of) coefficient values.



$$\overbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \underbrace{\sum_{j=1}^{p}\beta_j x_{ij}}_{\text{Predicted value}}\right)^2}^{\text{RSS}} + \lambda \overbrace{\sum_{j=1}^{p}\boxed{\beta_j^2}}^{\text{Shrinkage penalty}}$$

Ridge regression

- For each value of λ, we only have to fit one model:

$$\overbrace{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)}^{\text{RSS}} + \underset{\text{Tuning parameter}}{\lambda}\overbrace{\sum_{j=1}^{p}\beta_j^2}^{\text{Shrinkage penalty}}$$

- Substantial computational savings over best subset!

Ridge regression

- Question: what happens when the tuning parameter is small?
- Answer: just minimizing RSS; simple least-squares

$$\overbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)}^{\text{RSS}} + \underset{\substack{\nearrow \\ \text{Tuning} \\ \text{parameter}}}{\lambda} \overbrace{\sum_{j=1}^{p}\beta_j^2}^{\substack{\text{Shrinkage} \\ \text{penalty}}}$$

- Question: what happens when the tuning parameter is large?
- Answer: all coefficients go to zero; turns into null model

Ridge regression

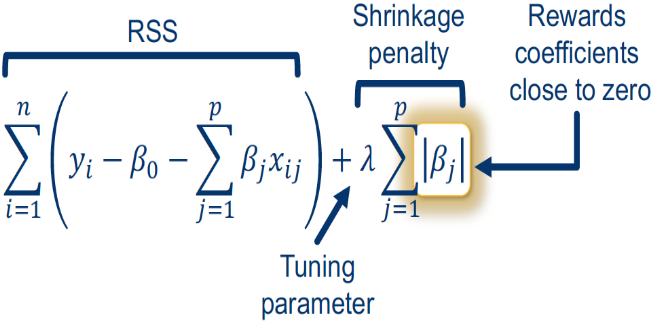
- Question: why would ridge regression improve the fit over least-squares regression?
- Answer: comes down to bias-variance tradeoff
  - As λ increases, flexibility decreases: variance decreases and bias increases
  - As λ decreases, flexibility increases: variance increases and bias decreases
- Takeaway: ridge regression works best in situations where least squares estimates have high variance:
  - trades a small increase in bias for a large reduction in variance

Ridge regression

- Ridge regression doesn't actually perform variable selection
- Final model will include all predictors
  - If all we care about is prediction accuracy, this isn't a problem
  - It does, however, pose a challenge for model interpretation

The Lasso

- The lasso is an alternative to ridge regression that overcomes the problem of including all p predictors in the final model.
- Big idea: minimize RSS plus an additional penalty that rewards small (sum of) coefficient values



$$\overbrace{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)}^{\text{RSS}}+\lambda\overbrace{\sum_{j=1}^{p}\boxed{|\beta_j|}}^{\text{Shrinkage penalty}}$$

RSS

Shrinkage penalty

Rewards coefficients close to zero

Tuning parameter

The Lasso

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.
- The lasso penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
  - Hence, much like best subset selection, the lasso performs variable selection.
- Models generated from the lasso are generally much easier to interpret than those produced by ridge regression.
- The lasso yields sparse models – that is, models that involve only a subset of the variables.

Comparing ridge regression and the lasso

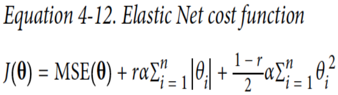
- Both significantly reduce variance at the expense of a small increase in bias.
- Question: when would one outperform the other?
- Answer:
  - When there are relatively many equally-important predictors, ridge regression will dominate
  - When there are small number of important predictors and many others that are not useful, the lasso will win

Selecting the Tuning Parameter

- Question: how do we choose the right value of λ?
- Answer: Cross validation
  - We choose a grid of λ values, and compute the cross-validation error for each value of λ
  - We then select the tuning parameter value for which the cross-validation error is smallest. Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter
- The model is re-fit using all of the available observations and the selected value of the tuning parameter.

Elastic Net

- Elastic Net is a middle ground between Ridge Regression and Lasso Regression.
- The regularization term is a simple mix of both Ridge and Lasso's regularization terms, and control the mix ratio r.



Equation 4-12. Elastic Net cost function

$$J(\boldsymbol{\theta}) = \mathrm{MSE}(\boldsymbol{\theta}) + r\alpha\Sigma_{i=1}^{n}\left|\theta_i\right| + \frac{1-r}{2}\alpha\Sigma_{i=1}^{n}\theta_i^2$$

- When $r = 0$, Elastic Net is equivalent to Ridge Regression, and
- When $r = 1$, it is equivalent to Lasso Regression.

Summary

- So when should you use plain Linear Regression (i.e., without any regularization), Ridge, Lasso, or Elastic Net?
- Generally you should avoid plain Linear Regression.
- Ridge is a good default.
  - However, if you suspect that only a few features are useful, you should prefer Lasso or Elastic Net since they tend to reduce the useless features' weights down to zero.
- In general, Elastic Net is preferred over Lasso because:
  - Lasso may behave erratically when the number of features is greater than the number of training instances or
  - When several features are strongly correlated.