THE UNIVERSITY OF SYDNEY

Lecture 9: Feature (model) selection

STAT5003

Pengyi Yang

Why select features (models)?

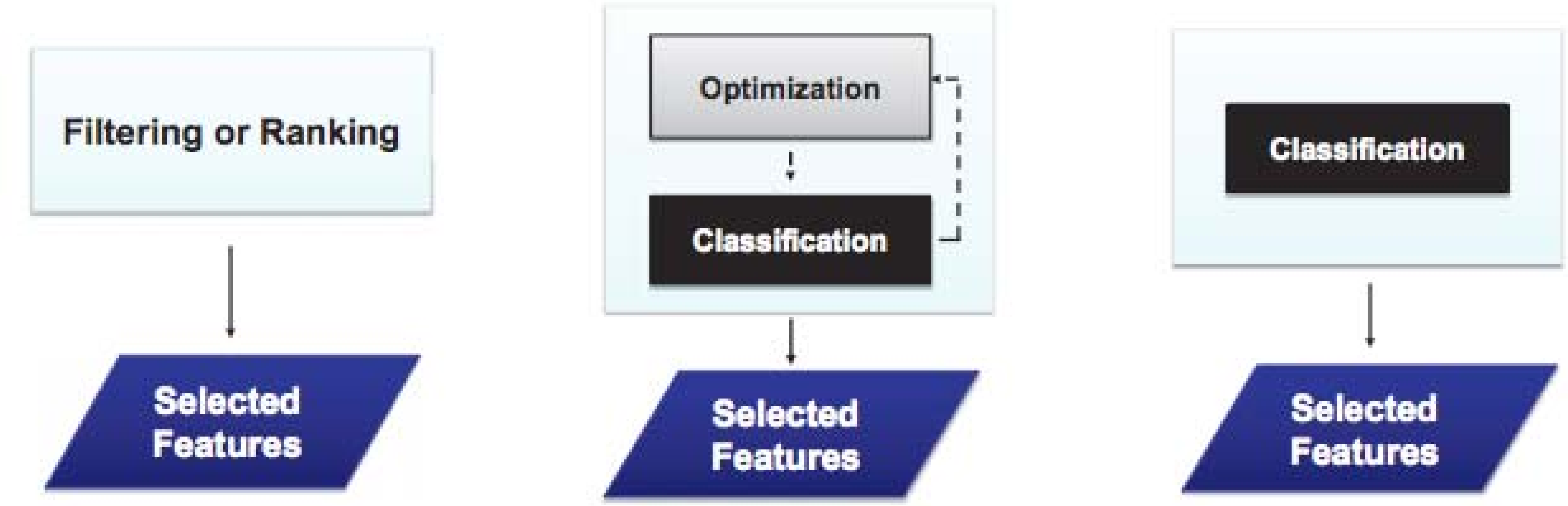
- *Prediction accuracy:* especially when $p > n$ (p denotes the number of descriptive features and n denotes number of samples), to control the variance of the model.

- *Model interpretability:* by removing irrelevant features (that is, by setting the corresponding coefficient estimates to zero), we can obtain a model that is more easily interpreted. We will present various approaches for feature selection.

Three classes of methods

1. *Subset selection*. We identify a subset of the *p* predictors that we believe to be related to the response or class (*y*). We than fit a classification or regression model on the reduced set of variables.
2. *Shrinkage*. This is primarily used for regression models in that we fit a model involving all *p* predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as regularisation) has the effect of reducing variance and can also used for feature selection.
3. *Dimension reduction*. We project the *p* predictors into *M*-dimensional subspace, where $M < p$. This is achieved by computing M different *linear combinations*, or *projections*, of the variables. Then these M projections are used as predictors to fit a classification or regression model.

Subset selection techniques



Filtering or Ranking

Selected Features

(a) Filter approach

Optimization

Classification

Selected Features

(b) Wrapper approach

Classification

Selected Features

(c) Embedded approach

Filter

- Filter methods analyse intrinsic properties of data, ignoring the classifier. Most of these methods can perform two operations, ranking and subset selection: in the former, the importance of each individual feature is evaluated. In the latter, the final subset of features to be selected is provided. In some cases, these two operations are performed sequentially (first the ranking, then the selection); in other cases, only the selection is carried out. Filter methods suppress the least interesting features. These methods are particularly effective in computation time and robust to overfitting.
- However, filter methods tend to select redundant features because they do not consider the relationships between features. Therefore, they are mainly used as a pre-process method.

Fold Change

- Fold change between classes can be used as a simple filter method.
- It involves calculating the mean values within the class and then the ratio of the means between the classes.
- The absolute log2 ratios of the means between the classes are often taken for sorting the features from the “most changed” to the “least changed”.
- A cutoff is applied to select a subset of features based on the absolute fold change.

Demonstration

Statistical test of significance between groups

- The issue with fold change is that it does not taken into account the variability within the class. Without considering variability within the class, outliers can cause large fold change for a feature that are not informative for discriminating classes.
- A more sensible way is to use statistics test (such as t-test) that can take into account variability within each class.
- In this way, features can be ranked and selected according to their test statistics or corresponding *p*-values.

Demonstration

Wrapper

Best subset selection

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently classification error
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, or RSS.

More about best subset selection

- For computational reasons, best subset selection cannot be applied with very large *p*. *Why not?*

- Best subset selection may also suffer from statistical problems when *p* is large: larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data. Thus an enormous search space can lead to *overfitting* and high variance of the coefficient estimates.

- For both these reasons, *stepwise* methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

Forward stepwise 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.

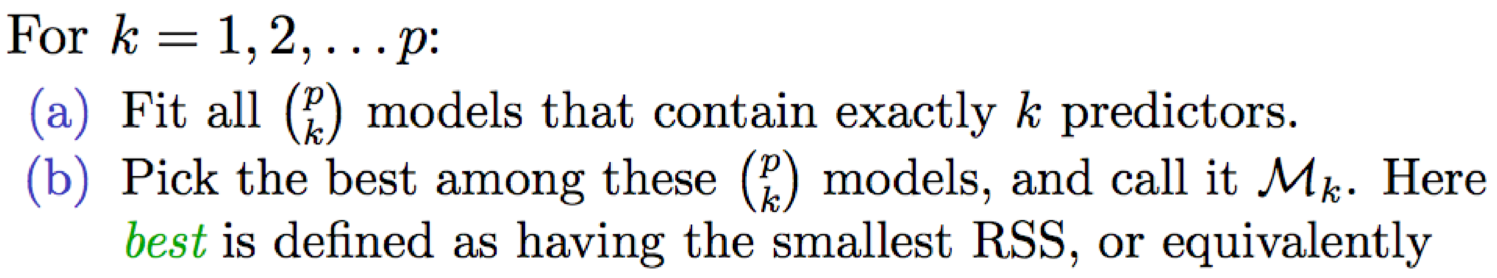
- In particular, at each step the variable that gives the greatest *additional* improvement to the fit is added to the model.

Forward stepwise selection in detail

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   - 2.1 Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   - 2.2 Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or classification error
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, or RSS.

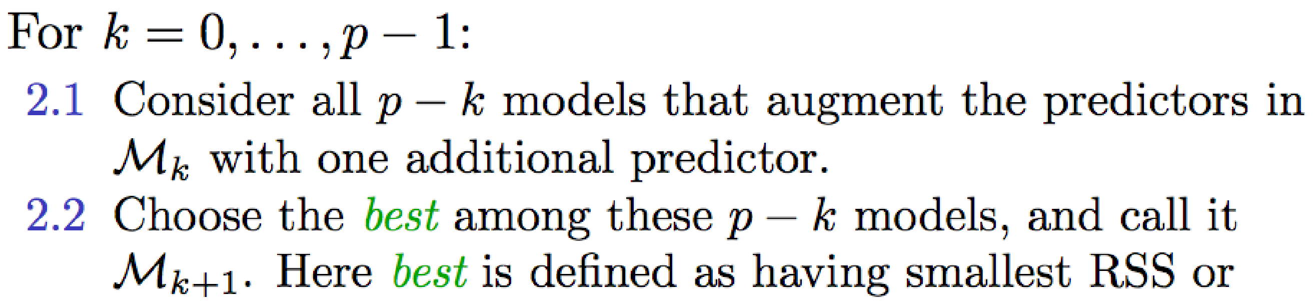
More on forward stepwise selection

- Computational advantage over best subset selection is clear.



```
For $k = 1, 2, \ldots p$:
  (a) Fit all $\binom{p}{k}$ models that contain exactly $k$ predictors.
  (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here
      *best* is defined as having the smallest RSS, or equivalently
      classification error
```

Best subset



```
For $k = 0, \ldots, p - 1$:
  2.1 Consider all $p - k$ models that augment the predictors in
      $\mathcal{M}_k$ with one additional predictor.
  2.2 Choose the *best* among these $p - k$ models, and call it
      $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or
      classification error
```

Forward stepwise

- It is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors. *Why not?*

Demonstration

Backward stepwise selection

- Like forward stepwise selection, *backward stepwise selection* provides an efficient alternative to best subset selection.

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

Backward stepwise selection: details

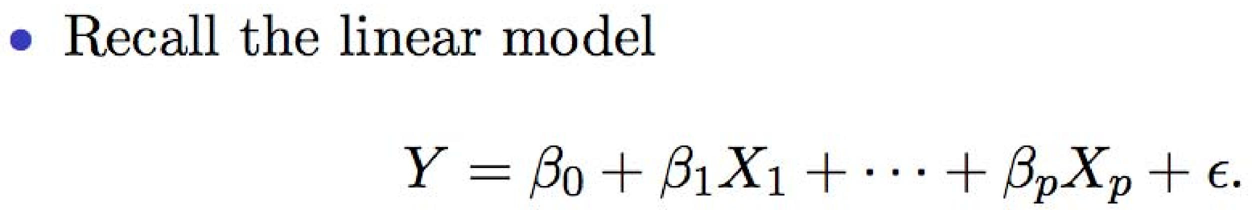
1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
    - 2.1 Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
    - 2.2 Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or classification error
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error

More on backward stepwise selection

- Like forward stepwise selection 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
- Like forward stepwise selection, backward selection is not guaranteed to yield the best model containing a subset of the p predictors.
- Note: for some models such as linear regression, backward selectin requires that the *number of samples n is larger than the number of features p* (so that the full model can be fit). In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset selection method (so far) when p is very large.

Details on evaluation for model selection

Linear model (feature) selection



- Recall the linear model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon.$$

How to choosing the optimal model?

- The model containing all of the predictors will always have the smallest RSS, since these quantities are related to the training error.
- We wish to choose a model with low test error, not a model with low training error. Recall that training error is usually a poor estimate of test error.
- Therefore, RSS are not suitable for selecting the best model among a collection of models with different number of predictors.

Estimating test error: two approaches

- We can *indirectly* estimate test error by making an adjustment to the training error to account for the bias due to overfitting.

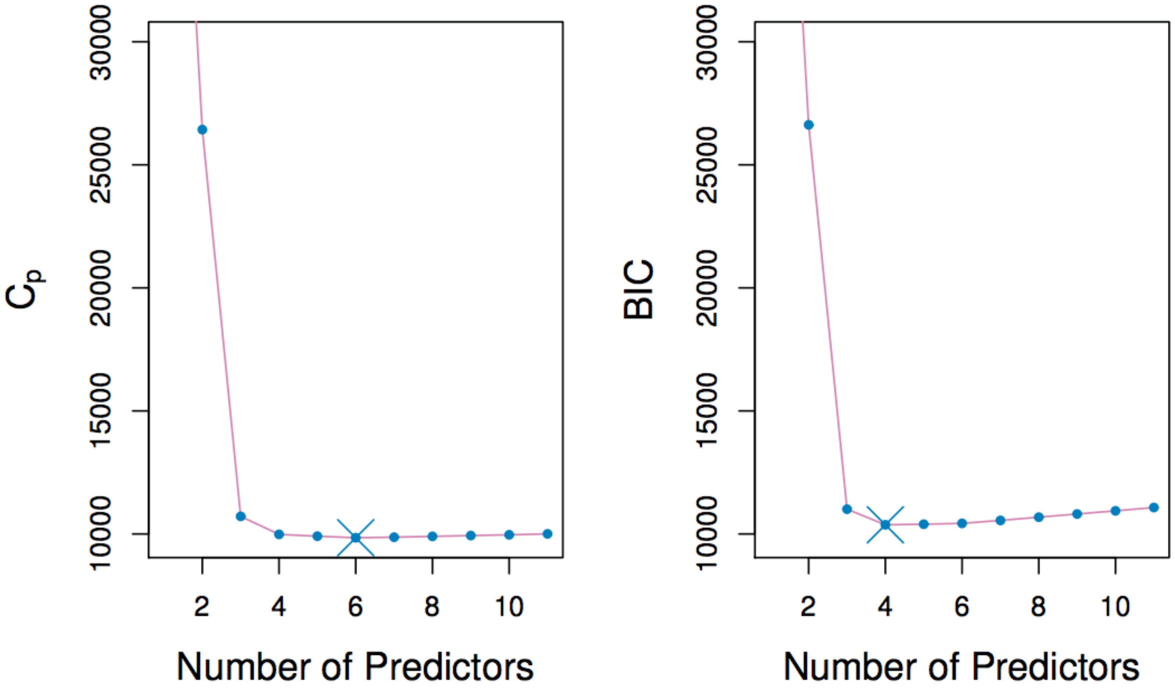
- We can *directly* estimate the test error, using either a validation set approach or a cross-validation approach, as discussed in previous lectures.

- We will illustrate *indirect* approach and also review cross-validation approach next.

Indirect approaches (C_p and BIC)

- These techniques adjust the training error for the model size (model complexity), and can be used to select among a set of models with different numbers of features.

- The figure below displays Mallow's CP (C_p) and BIC for the best model produced by best subset selection on the Credit data set (in book “An introduction to statistical learning”).



Details of C_p and BIC

Mallow's C_p

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

where d is the total # of parameters used and $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ associated with each response measurement.

$$\text{BIC} = \frac{1}{n}\left(\text{RSS} + \log(n)d\hat{\sigma}^2\right).$$

- Like C_p, the BIC will tend to take on small value for model with a low test error, and so generally we select model that has the lowest BIC value.
- Notice that BIC replaces the $2d\hat{\sigma}^2$ used by C_p with a $\log(n)d\hat{\sigma}^2$ term, where n is the number of samples.
- Since $\log n > 2$ for any $n > 7$, the BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than C_p.

Validation set and cross-validation

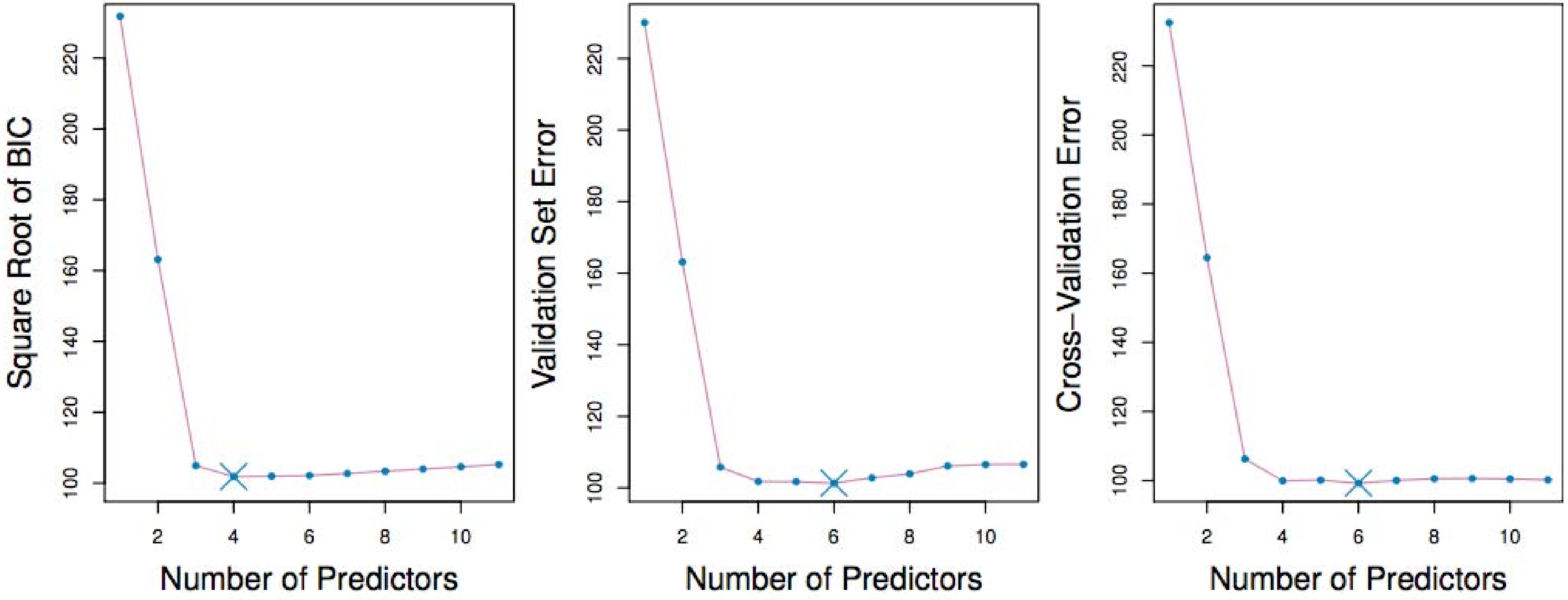
- Each of the procedures returns a sequence of models $\mathcal{M}_k$ indexed by model size $k = 0, 1, 2, \ldots$ Our job here is to select k. Once selected, we will return model $\mathcal{M}_k$

- We compute the validation set error or the cross-validation error for each model $\mathcal{M}_k$ under consideration, and then select the k for which the resulting estimated test error is smallest.

- This procedure has an advantage relative to C_p and BIC, in that it provides direct estimate of the test error, and *doesn't require an estimate of the error variance σ^2*.

- It can also be used in a wider range of model selection tasks, even in cases where it is hard to pinpoint the model degrees of freedom (e.g. the number of predictors in the model) or hard to estimate the error variance σ^2.

Credit data example



Shrinkage methods

We will introduce two methods specifically designed for linear regression.

Ridge regression and *Lasso*

- The subset selection methods use least squares to fit a linear model that contains a subset of the predictors.
- As an alternative, we can fit a model containing all *p* predictors using a technique that *constrains* or *regularises* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero.
- It may not be immediately obvious why such a constraint should improve the fit, but it turns out that shrinking the coefficient estimates can significantly reduce their variance.

Ridge regression

- Recall that the least squares fitting procedure estimates $\beta_0, \beta_1, \ldots, \beta_p$ using the values that minimize

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

- In contrast, the ridge regression coefficient estimates $\hat{\beta}^R$ are the values that minimize

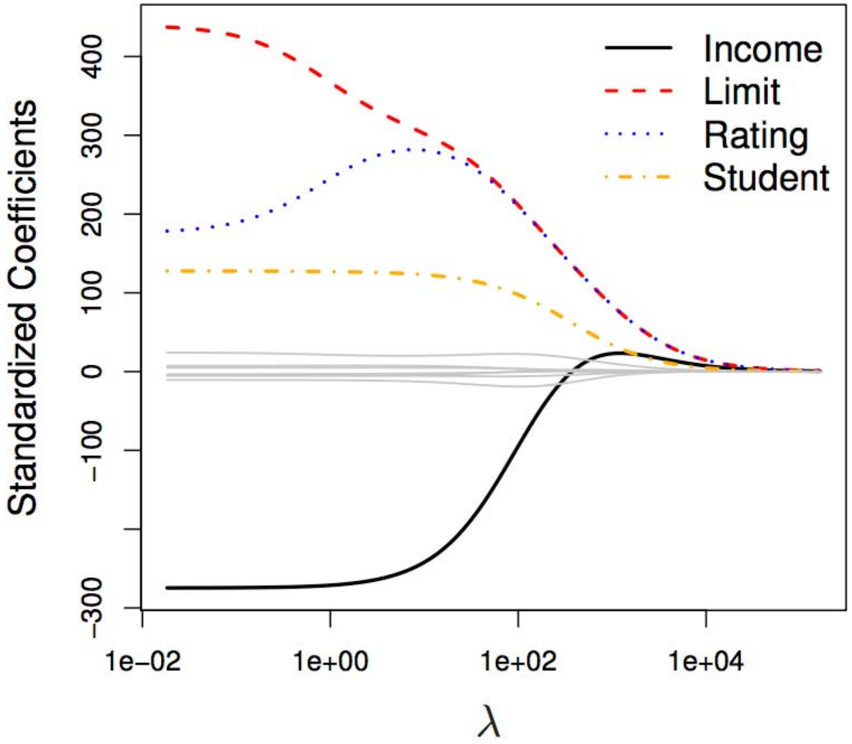
$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2 ,$$

where $\lambda \geq 0$ is a *tuning parameter*, to be determined separately.

Ridge regression: continued

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda \sum_j \beta_j^2$, called a *shrinkage penalty*, is small when $\beta_1, \dots, \beta_p$ are close to zero, and so it has the effect of *shrinking* the estimates of β_j towards zero.
- The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.
- Selecting a good value for λ is critical; cross-validation is used for this.

Credit data example



- Each curve corresponds to the ridge regression coefficient estimate for one of the ten features, plotted as a function of λ.

Demonstration

Ridge regression: scaling of predictors

- The standard least squares coefficient estimates are scale *invariant*: multiplying X_j by a constant c simply leads to a scaling of the least squares coefficient estimates by a factor of $1/c$. In other words, regardless of how the *j*th predictor is scaled, $X_j\hat{\beta}_j$ will remain the same.
- In contrast, the ridge regression coefficients estimates can change *substantially* when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function.
- Therefore, it is best to apply ridge regression after *standardising the predictors*, using a formula such as below:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

The Lasso

- Ridge regression does have one obvious disadvantage: unlike subset selection, which will generally select models that involve just a subset of features, ridge regression will include all *p* predictors in the final model
- The Lasso is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients, $\hat{\beta}^L$, minimise the quantity

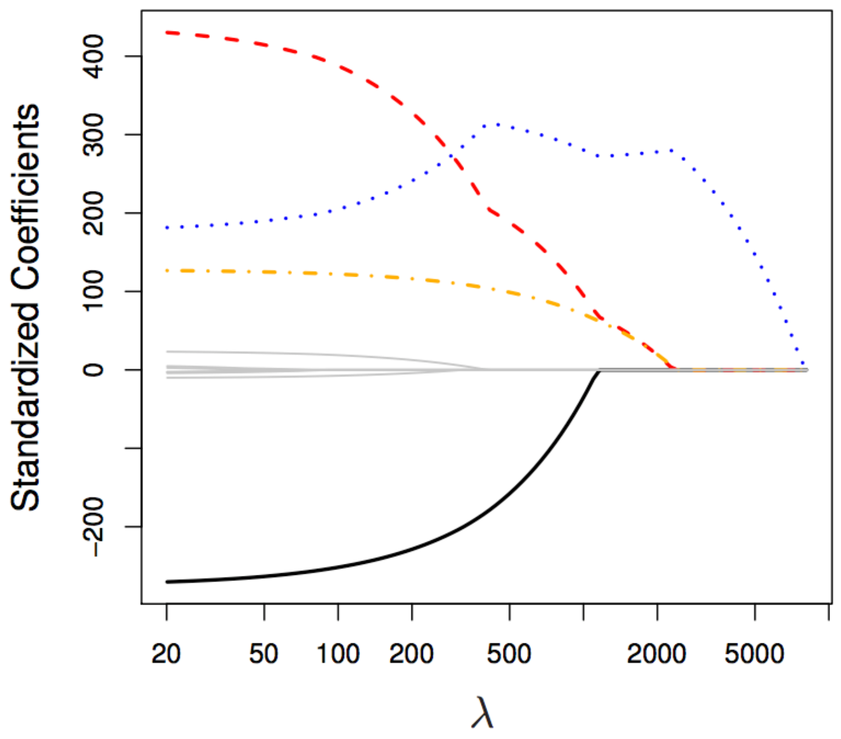
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

- The lasso uses an ℓ_1 penalty instead of the ℓ_2 penalty. The ℓ_1 norm of a coefficient vector β is given by $\| \beta \|_1 = \sum|\beta_j|$.

The Lasso: continued

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.
- However, in the case of the lasso, the ℓ_1 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 *feature selection* (in an embedded manner).
- We say that the lasso yields *sparse* models – that is, models that involve only a subset of variables.
- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.

Example: Credit dataset



Demonstration

The variable selection property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

One can show that the lasso and ridge regression coefficient estimates solve the problems

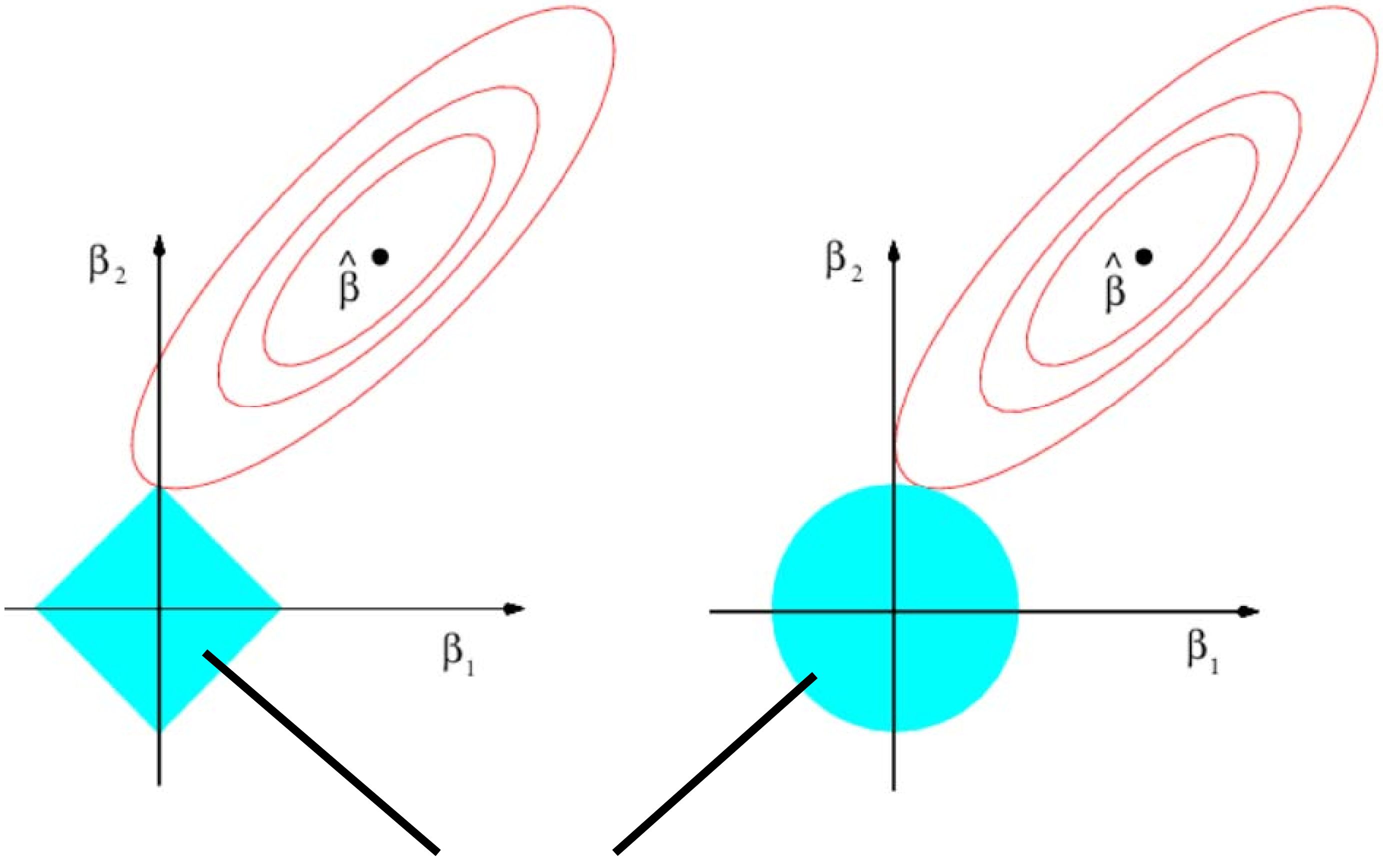
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

and

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

respectively.

Comparison of ℓ_1 and ℓ_2 constrains

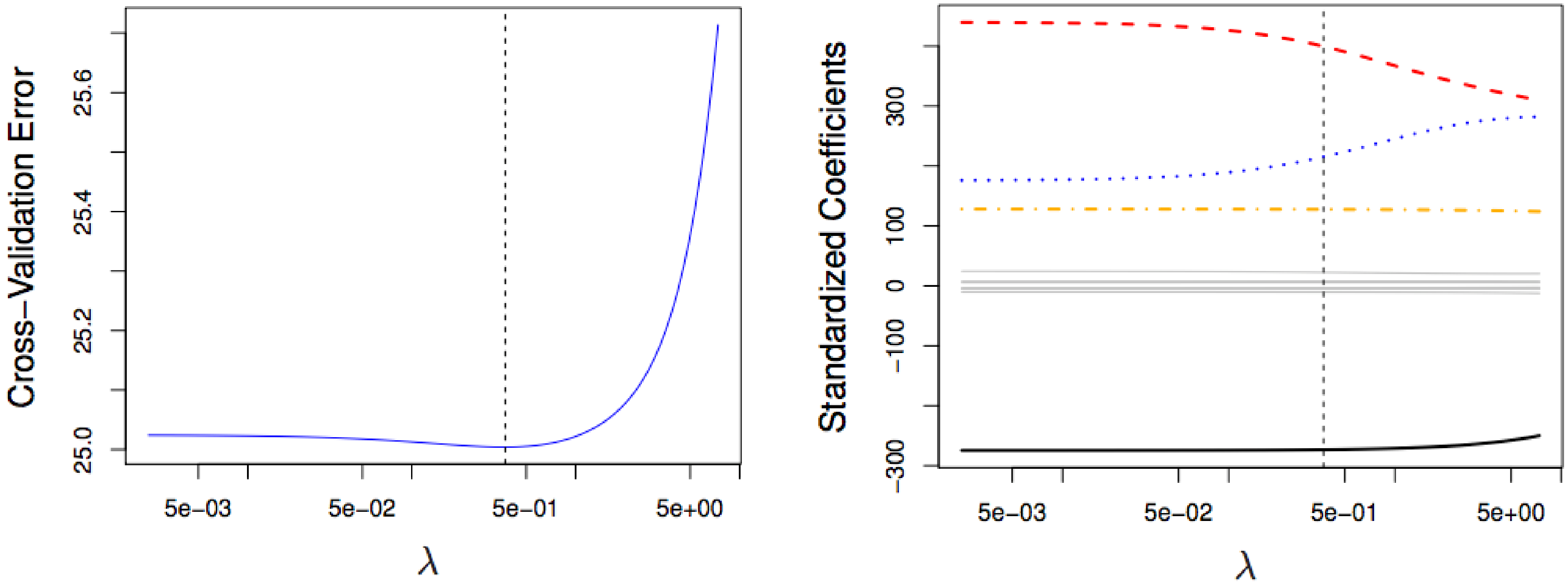


A solution is feasible if it is within this region

Selecting the tuning parameters for Ridge regression and Lasso

- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration is the best.
- That is, we require a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint *s*.
- *Cross-validation* provides a simple way to tackle this problem. We choose a grid of λ values, and compute the cross-validation error rate 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.

Credit data example



- Left illustrates cross-validation errors that result from applying ridge regression to the Credit data set with a range of λ values.
- Right illustrate the coefficient estimates as a function of λ. The vertical dashed lines indicate the best value of λ selected by cross-validation.