

# Foundation of Machine Learning (CSE4032)

## Lecture 04: Subset Selection and Shrinkage

**Dr. Kundan Kumar**  
Associate Professor  
Department of ECE



Faculty of Engineering (ITER)  
S'O'A Deemed to be University, Bhubaneswar, India-751030  
© 2021 Kundan Kumar, All Rights Reserved

# Outline

- 1 Introduction
- 2 Subset Selection
- 3 Shrinkage Methods
- 4 References

# Subset Selection

# Introduction

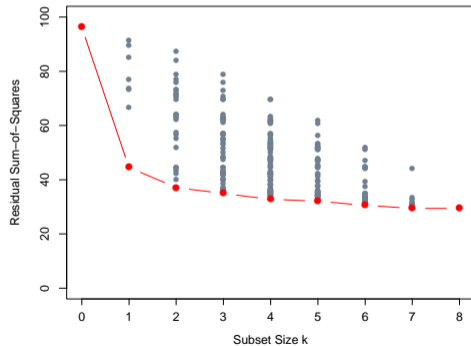
- **Variable subset selection** is a technique which is used for data reduction in data mining process. Data reduction reduces the size of data so that it can be used for analysis purposes more efficiently.
- **Benefits** of performing variable selection/reduction:
  - avoid curse of dimensionality
  - reduce the computational cost
  - improves accuracy
  - avoid overfitting
- Worth to mention here that we are often not satisfied with the least squares estimates
  - The first is **prediction accuracy**: the least squares estimates often have low bias but large variance.
  - The second reason is **interpretation**. With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects.

# Introduction

- Carefully selected variables can improve model accuracy. But adding too many can lead to overfitting.
- The first step in variable selection is to define a **criterion function** that is often a **function of the classification/residual error**.
- Note that, the use of classification/residual error in the criterion function makes variable selection procedures dependent on the specific model used.
- The most straightforward approach would require
  - (i) Examining all  $\binom{p}{k}$  or  ${}^pC_k$  possible subsets of size  $k$ ,
  - (ii) Selecting the subset that performs the best according to the criterion function.
- The number of subsets grows combinatorially, making the **exhaustive search** impractical.

# Best-Subset Selection

- Best subset regression finds for each  $k \in \{0, 1, 2, \dots\}$  the subset of size  $k$  that gives smallest residual sum of squares (RSS).
- An efficient algorithm—the **leaps and bounds procedure** (Furnival and Wilson, 1974) — makes this feasible for  $p$  as large as 30 or 40.
- The question of how to choose  $k$  involves the tradeoff between bias and variance, typically we choose the smallest model that minimizes an estimate of the expected prediction error.



# Stepwise selection

- An exhaustive search for the subset may not be feasible if  $p$  is very large.
- There are two main alternatives.
  - Forward stepwise selection
  - Backward stepwise selection
- **Forward stepwise selection:**
  - First, we approximate the response variable  $Y$  with a constant (i.e., an intercept-only regression model).
  - Then, we gradually add one more variable at a time (or add main effects first, then interactions).
  - Every time we always choose from the rest of the variables the one that yields the best accuracy in prediction when added to the pool of already selected variables. This accuracy can be measured by the R-square, F-statistic, AIC, BIC, etc.

# Stepwise selection

- For example, if we have 10 predictor variables, first we would approximate  $Y$  with a constant, and then use one variable out of the 10 (I would perform 10 regressions, each time using a different predictor variable; for every regression I have a residual sum of squares; the variable that yields the minimum residual sum of squares is chosen and put in the pool of selected variables). We then proceed to choose the next variable from the 9 left, etc.
- **Backward stepwise selection:**
  - This is similar to forward stepwise selection, except that we start with the full model using all the predictors and gradually delete variables one at a time.
  - There are various methods developed to choose the number of predictors, for instance the F-ratio test. We stop forward or backward stepwise selection when no predictor produces an F-ratio statistic greater than some threshold.



# Examples: Iris data representation ( $k = 1$ )

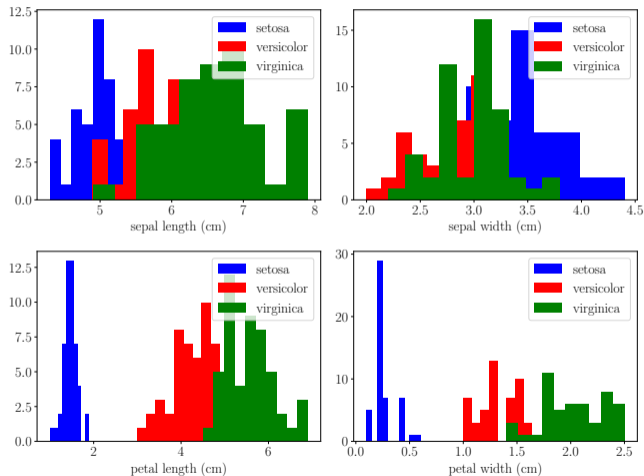


Figure: Histogram plot of Iris features

# Examples: Iris data representation ( $k = 2$ )

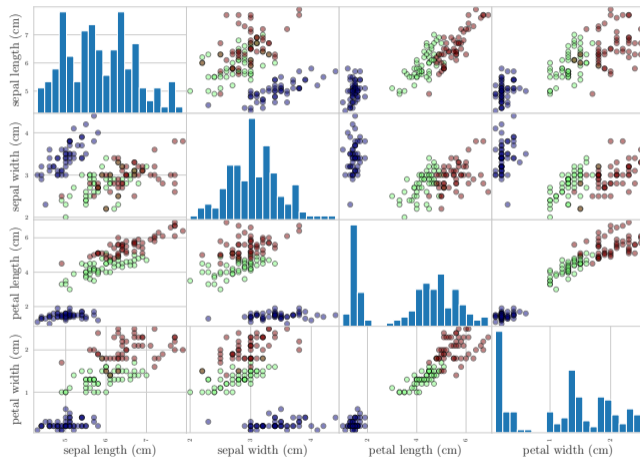
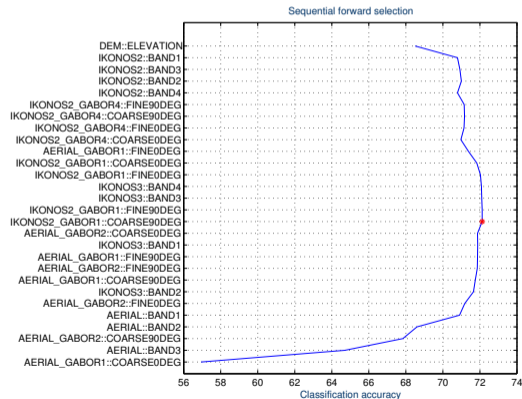


Figure: Scatter plot of the iris data. Off-diagonal cells show scatters of pairs of features  $x_1, x_2, x_3, x_4$ .

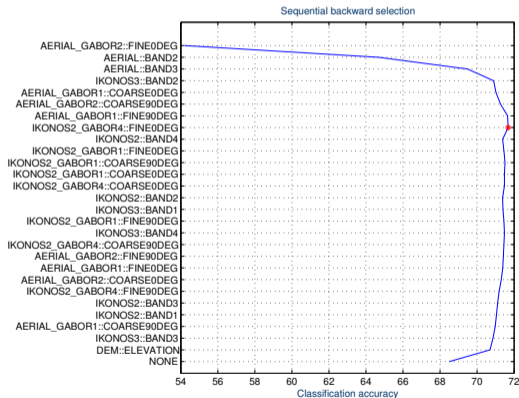
# Example: Stepwise forward selection

1. First, the best single feature is selected.
2. Then, pairs of features are formed using one of the remaining features and this best feature, and the best pair is selected.
3. Next, triplets of features are formed using one of the remaining features and these two best features, and the best triplet is selected.
4. This procedure continues until all or a predefined number of features are selected.



# Example: Stepwise backward selection

- First, the criterion function is computed for all  $p$  features.
- Then, each feature is deleted one at a time, the criterion function is computed for all subsets with  $p - 1$  features, and the worst feature is discarded.
- Next, each feature among the remaining  $p - 1$  is deleted one at a time, and the worst feature is discarded to form a subset with  $p - 2$  features.
- This procedure continues until one feature or a predefined number of features are left.



# Summary

- The choice between dimensionality reduction and variable selection depends on the application domain and the specific training data.
- Variable selection leads to savings in computational costs and the selected variables retain their original physical interpretation.
- Dimensionality reduction with transformations may provide a better discriminative ability but these new variables may not have a clear physical meaning.
- There is no guarantee that the subsets obtained from stepwise procedures will contain the same variables or even be the “best” subset.
- When there are more variables than observations ( $p > n$ ), backward elimination is typically not a feasible procedure.

# Shrinkage Methods

# Introduction

- It is not unusual to see the number of input variables greatly exceed the number of observations, e.g. micro-array data analysis, environmental pollution studies.
- Subset selection is a discrete process—variables are either retained or discarded—it often exhibits high variance, and so doesn't reduce the prediction error of the full model.
- Shrinkage methods are more continuous, and don't suffer as much from high variability.
- Shrinkage methods for regression
  - Ridge Regression
  - The Lasso

# Ridge Regression

- Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where **independent variables are highly correlated**.
- With many predictors, fitting the full model without penalization will result in large prediction intervals, and LS regression estimator may not uniquely exist.
- Because the LS estimates depend upon  $(\mathbf{X}^T \mathbf{X})^{-1}$ , we would have problems in computing  $\beta_{LS}$  if  $\mathbf{X}^T \mathbf{X}$  were singular or nearly singular.
- In those cases, small changes to the elements of  $X$  lead to large changes in  $(\mathbf{X}^T \mathbf{X})^{-1}$ .
- The least square estimator  $\beta_{LS}$  may provide a good fit to the training data, but it will **not fit sufficiently well to the test data**.
- One way out of this situation is to abandon the requirement of an unbiased estimator.



# Ridge Regression

- We assume only that  $\mathbf{X}$ 's and  $\mathbf{y}$  have been centered, so that we have no need for a constant term in the regression:
  - $\mathbf{X}$  is a  $n \times (p)$  matrix with centered columns,
  - $\mathbf{y}$  is a centered  $n$ -vector.
- Hoerl and Kennard (1970) proposed that potential instability in the LS estimator

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

could be improved by adding a small constant value  $\lambda$  to the diagonal entries of the matrix  $\mathbf{X}^T \mathbf{X}$  before taking its inverse. The result is the ridge regression estimator

$$\hat{\beta}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^T \mathbf{y}$$

# Ridge Regression

- Ridge regression shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares,

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}.$$

Here  $\lambda \geq 0$  is a complexity parameter that controls the amount of shrinkage: the larger the value of  $\lambda$ , the greater the amount of shrinkage.

- Which is equivalent to

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2,$$

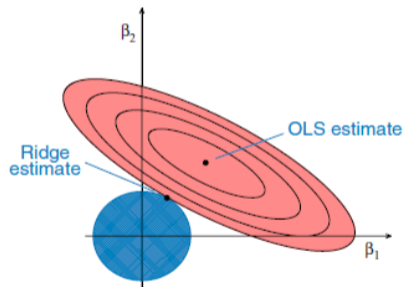
subject to  $\sum_{j=1}^p \beta_j^2 \leq t.$

# Ridge Regression

- Therefore, ridge regression puts further constraints on the parameters,  $\beta_j$ 's, in the linear model.
- In this case, what we are doing is that instead of just minimizing the residual sum of squares we also have a penalty term on the  $\beta$ 's. This penalty term is  $\lambda$  (a **pre-chosen constant**) times the squared norm of the  $\beta$  vector.
- This means that if the  $\beta_j$ 's take on large values, the optimization function is penalized.
- We would prefer to take smaller  $\beta_j$ 's, or  $\beta_j$ 's that are close to zero to drive the penalty term small.

# Geometric Interpretation of Ridge Regression

- The ellipses correspond to the contours of residual sum of squares (RSS): the inner ellipse has smaller RSS, and RSS is minimized at ordinary least square (OLS) estimates.
- For  $p = 2$ , the constraint in ridge regression corresponds to a circle,  $\sum_{j=1}^p \beta_j^2 < t$ .
- We are trying to minimize the ellipse size and circle simultaneously in the ridge regression. The ridge estimate is given by the point at which the ellipse and the circle touch.
- There is a trade-off between the penalty term and RSS.



# Geometric Interpretation of Ridge Regression

- Maybe a large  $\beta$  would give you a better residual sum of squares but then it will push the penalty term higher. This is why you might actually prefer smaller  $\beta$ 's with worse residual sum of squares.
- From an optimization perspective, the penalty term is equivalent to a constraint on the  $\beta$ 's.
- The function is still the residual sum of squares but now you constrain the norm of the  $\beta_j$ 's to be smaller than some constant  $t$ .
- There is a correspondence between  $\lambda$  and  $t$ . The larger the  $\lambda$  is, the more you prefer the  $\beta_j$ 's close to zero. In the extreme case when  $\lambda = 0$ , then you would simply be doing a normal linear regression.
- And the other extreme as  $\lambda$  approaches infinity, you set all the  $\beta$ 's to zero.

# The Lasso

# Lasso

- The lasso is a shrinkage method like ridge, with subtle but **important differences**. The lasso estimate is defined by

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2$$

subject to  $\sum_{j=1}^p |\beta_j| \leq t.$

- Just as in ridge regression, we can re-parametrize the constant  $\beta_0$  by standardizing the predictors; the solution for  $\hat{\beta}_0$  is  $\bar{y}$ , and thereafter we fit a model without an intercept.
- In the signal processing literature, the lasso is also known as **basis pursuit** (Chen et al., 1998).

# Lasso

- We can also write the lasso problem in the equivalent Lagrangian form

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}.$$

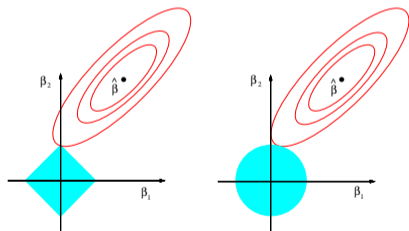
Notice the similarity to the ridge regression problem: the  $L_2$  ridge penalty  $\sum_1^p \beta_j^2$  is replaced by the  $L_1$  lasso penalty  $\sum_1^p |\beta_j|$ .

- Some of the coefficients may be shrunk exactly to zero. The least absolute shrinkage and selection operator, or lasso, as described in Tibshirani (1996) is a technique that has received a great deal of interest.





# Geometric Interpretation

- The lasso performs L1 shrinkage, so that there are “corners” in the constraint, which in two dimensions corresponds to a diamond. If the sum of squares “hits” one of these corners, then the coefficient corresponding to the axis is shrunk to zero.
- As  $p$  increases, the multidimensional diamond has an increasing number of corners, and so it is highly likely that some coefficients will be set equal to zero. Hence, the lasso performs shrinkage and (effectively) subset selection.
- In contrast with subset selection, Lasso performs a soft thresholding: as the smoothing parameter is varied, the sample path of the estimates moves continuously to zero.



# References

-  The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Hastie, Tibshirani, and Friedman, Springer.
-  In Introduction to Statistical Learning with Application in R, Second Edition, James, Witten, Hastie, and Tibshirani, Springer.



*Thank you!*