Department of Statistics
Lund University

Data Mining
Fall

Project 5: Variance reduction, lasso method, neural networks

Peform all requested task. Some useful R-code that can help in completing Project 5 can be found through the Lab 5 link in our webpage.

Part One – Random Forest and variance reduction

In the lecture on random forest, it was argued that random forest by sampling coordinates can reduce variance of an estimator. The first part of the lab is presenting a Monte Carlo study that illustrates the variance reduction for heavily correlated bivariate data. The following represent bivariate data that are correlated both between the two variables as well as between samples. The data constitute a $N \times 2$ matrix

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} \\ \vdots & \vdots \\ X_{N1} & X_{N2} \end{bmatrix}$$

We want them to be correlated both between rows and between columns. The correlations for matrices of random variables are often presented through the covariance matrix that is obtained for the vector obtained from the matrix $\mathbf{X}$ by stacking columns one at the top of the other which is denoted by $vec(\mathbf{X})$.

$$vec(\mathbf{X}) = \begin{bmatrix} X_{11} \\ \vdots \\ X_{N1} \\ X_{12} \\ \vdots \\ X_{N2} \end{bmatrix}$$

Let $\mathbf{Z}$, $\mathbf{Z}_N$, $\mathbf{Z}_2$ be $N \times 2$, $N \times 1$, and 1×2 matrices of iid standard normal variables. Let $\mathbf{1}_{.2}$ be two dimensional row of ones and $\mathbf{1}_{N.}$ be N dimensional column of ones. We define our data through

$$\mathbf{X} = \sqrt{1-\rho^2}\sqrt{1-\rho_0^2}\mathbf{Z} + \rho_0 \mathbf{Z}_N \mathbf{1}_{.2} + \rho \mathbf{1}_{N.} \mathbf{Z}_2.$$

One can see that ρ_0 introduces correlation between columns in $\mathbf{X}$ and ρ between rows.

(1) Simulated data from the data, choose Monte Carlo sample of size 2000 and evaluate covariance of the stacked vector. Does the estimate of the covariance resembles the

theoretical covariance that is given in

$$Cov(\mathbf{X}) = \begin{bmatrix} \begin{bmatrix} 1 & \rho^2 & \dots & \rho^2 \\ \vdots & \vdots & \vdots & \vdots \\ \rho^2 & \rho^2 & \dots & 1 \end{bmatrix} & \begin{bmatrix} \rho_0^2\sqrt{1-\rho^2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \rho_0^2\sqrt{1-\rho^2} \end{bmatrix} \\ \begin{bmatrix} \rho_0^2\sqrt{1-\rho^2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \rho_0^2\sqrt{1-\rho^2} \end{bmatrix} & \begin{bmatrix} 1 & \rho^2 & \dots & \rho^2 \\ \vdots & \vdots & \vdots & \vdots \\ \rho^2 & \rho^2 & \dots & 1 \end{bmatrix} \end{bmatrix}.$$

Comment on observed correlation within rows (between columns) in relation to correlation between rows. Where in the estimated covariance matrix you find the values that allow you to make a claim on that?

(2) Estimate the common mean of the coordinates of $\mathbf{X}$ using only a single sample of X. What is the true value of the mean? How many data have been use to get this estimated value? Is the result somewhat surprising?
(3) Perform standard boostrap of sampling pairs of the data to get a bootstrap estimate of the mean. Is it improving estimation?
(4) By sampling randomly coordinates perform random forest style bootstrap. Evaluate the estimate. Is it improving estimation?
(5) Perform a Monte Carlo study to see if the observed improvement is indeed meaningful. What is your conclusion about the random forest approach?

Part Two – Lasso method

We have seen that by using a penalty one can fit a regression model involving penalizing the coefficient estimates. Those methods often shrink the coefficient estimates towards zero. One of such methods is the lasso. The lasso coefficients, $\hat{\beta}_\lambda^L$ minimize the quantity

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

It turns out that shrinking the coefficient estimates can significantly reduce their variance. We will use the `glmnet` package in order to perform the lasso. The main function in this package is `glmnet()`, which can be used to fit lasso models. This function has slightly different syntax from other model-fitting functions. In particular, we must pass in an `x` matrix as well as a `y` vector, and one do not use the `y  x` syntax. In this lab the lasso is used in order to predict *Salary on the Hitters data*.

(1) Install and make yourself familiar with the package `glmnet()`.
(2) The explore the content of *Salary on the Hitters data*. Before proceeding ensure that the missing values have been removed from the data. *Salary* is missing for 59 players. The `na.omit()` function removes all of the rows that have missing values in any variable.
(3) Associated with each value of the penalty λ is a vector of ridge regression coefficients, stored in a matrix that can be accessed by `coef()`. In our case, it is a 20100 matrix, with 20 rows (one for each predictor, plus an intercept) and 100 columns (one for each value of λ).
(4) Interpret the coefficient plot, see how depending on the choice of tuning parameter, some of the coefficients will be exactly equal to zero.

(5) Perform cross-validation and compute the associated test error.

Part Three – Neural networks

This parts is a very basic introduction to building neural networks. We use neural network to 'learn' how to compute square root of a number.

(1) Load `Neuralnet` R-package. Type `?neuralnet` for more information on the neuralnet library.
(2) Generate the data: numbers from 0 to 100 on which the neural network will be trained to evaluate square root.
(3) Perform the learning and testing phase.
(4) Interpret the neural network presented on the graph.