PSet 4: Biased Methods and Regularization

Stat 154, Spring 2018

Due date: Fri Mar-16 (before midnight)

1) Properties of PLS Regression (20 pts)

Refer to the "classic" algorithm for PLS Regression

Set $\mathbf{X_0} = \mathbf{X}$ and $\mathbf{y_0} = \mathbf{y}$
for $h = 1, 2, \ldots, r$ **do**
  $\mathbf{w_h} = \mathbf{X_{h-1}^\top y_{h-1}}$
  normalize weights: $\|\mathbf{w_h}\| = 1$
  $\mathbf{z_h} = \mathbf{X_{h-1} w_h} / \mathbf{w_h^\top w_h}$
  $\mathbf{p_h} = \mathbf{X_{h-1}^\top z_h} / \mathbf{z_h^\top z_h}$
  $b_h = \mathbf{y_{h-1}^\top z_h} / \mathbf{z_h^\top z_h}$
  Deflations:
  $\mathbf{X_h} = \mathbf{X_{h-1}} - \mathbf{z_h p_h^\top}$
  $\mathbf{y_h} = \mathbf{y_{h-1}} - b_h \mathbf{z_h}$
end for

where r is the rank of $\mathbf{X}$

1.a) Weights and loadings are collinear (10 pts)

Prove that the inner product of weights and loadings is: $\mathbf{w}_h^\top \mathbf{p}_h = 1$

1.b) Weights and predictor-residuals are orthogonal (10 pts)

Here's another property of PLSR: $\mathbf{w_h^\top X_l^\top} = 0, l \geq h$. Prove this property for the case $l = h$.

2) Bias of Regression Coefficients in PCR (10 pts)

Assume that there is a linear relationship between the response and the predictors: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with $E(\boldsymbol{\varepsilon}) = \mathbf{0}$. PCR can be used to reduce the dimensionality of the regression by dropping those dimensions that contribute to the collinearity problem. The estimated regression coefficients $\mathbf{b}_Z^{(k)}$ for the k principal components ($\mathbf{Z}_k$) are given by:

$$\mathbf{b}_Z^{(k)} = (\mathbf{Z}_k^\top \mathbf{Z}_k)^{-1} \mathbf{Z}_k^\top \mathbf{y}$$

It is usual to transform the PCR coefficients $\mathbf{b}_Z^{(k)}$ into coefficients $\hat{\boldsymbol{\beta}}_Z^{(k)}$ of the original input variables:

$$\hat{\boldsymbol{\beta}}_Z^{(k)} = \mathbf{V}_k \mathbf{b}_Z^{(k)} \quad \Longrightarrow \quad \hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}_Z^{(k)}$$

where $\mathbf{V}_k$ is the matrix of k loadings.

2.a) Bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$

Show that the theoretical bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$ is:

$$E\left[\hat{\boldsymbol{\beta}}_Z^{(k)} - \boldsymbol{\beta}\right] = (\mathbf{V}_k \boldsymbol{\Lambda}_k^{-1} \mathbf{V}_k^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{X} - \mathbf{I})\boldsymbol{\beta}$$

- $\mathbf{V}_k$ is the matrix of k loadings associated to the k PCs (i.e. eigenvectors).
- $\boldsymbol{\Lambda}_k$ is the diagonal matrix of k PC variances (i.e. eigenvalues)

2.b) Bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$ when $k = p$

Suppose that $\mathbf{X}$ is of full column-rank p. What is the bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$ if all PCs are used in PCR, that is, if $k = p$?

3) Bias of Ridge Regression Coefficients (10 pts)

The mean squared error of the ridge regression estimator $\hat{\boldsymbol{\beta}}_r$ is given by:

$$\text{MSE}(k) = E\left[\left(\hat{\boldsymbol{\beta}}_r(k) - \boldsymbol{\beta}\right)^{\mathsf{T}} \left(\hat{\boldsymbol{\beta}}_r(k) - \boldsymbol{\beta}\right)\right] = Var(k) + Bias^2(k)$$

Using the SVD decomposition of $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$, show that the theoretical bias of the ridge estimator $\hat{\boldsymbol{\beta}}_r$ is:

$$E\left[\hat{\boldsymbol{\beta}}_r(k) - \boldsymbol{\beta}\right] = \{(\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathsf{T}} + k\mathbf{I})^{-1}\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathsf{T}} - \mathbf{I}\}\boldsymbol{\beta}$$

4) Models for Solubility Data

The next problems have to do with APM's Chapter 6: *Linear Regression and Its Cousins* (Kuhn and Johnson, 2013). The main idea is to apply PCR, PLSR, Ridge Regression, and Lasso, on the `solubility` data set.

Warning: Because the model fitting process requires using 10-fold CV, and tuning parameters, some computations will take some time. We recommend that you use an `R` script file (not to confuse with an `Rmd` file) to write code and experiment with it. Once you have the right commands, then you can include them in your `Rmd` file. Experimenting directly on the `Rmd` will cause the knitting process to run very slowly.

The data `solubility` comes in the R package `"AppliedPredictiveModeling"`. A description about the background for this data set is described in section 6.1 (page 102). The set of objects contained in `solubility` is described in section 6.5 (page 128).

Related functions, code examples, and outputs are described in section 6.5. The associated R packages are:

- `"AppliedPredictiveModeling"`
- `"caret"`
- `"pls"`
- `"elasticnet"`

Use the function `trainControl()` to specify 10-fold cross-validation as the type of resampling method:

```
# example on page 130
ctrl <- trainControl(method = "cv", number = 10)
```

Learn how to use the function `train()`—with 10-fold cross-validation—to fit models for: PCR, PLSR, Ridge Regression, and Lasso. An example of `train()` for PLSR is on page 134. Here's a list of argument values that you should use to fit the models via `train()`:

- use `solTrainXtrans` for the argument `x`
- use `solTrainY` for the argument `y`
- specify mean-centered and scaled data for the argument `preProcess`
- use `ctrl` for the argument `trControl`

Depending on which method you are using, you will have to modify some of the `train()` arguments such as: `method`, `tuneLength`, and `tuneGrid`.

4.1) PCR (20 pts)

- Set a random seed.
- Use `train()` to perform the model building process for PCR.
- Use `train()`'s argument `tuneLength = 40`.

- Store the `train()` output in an object called `pcr_fit`, and print this object.
- Plot the RMSEs against the number of PCs; use dots connected by a line (see figure 6.11, page 117 for an example of this type of plot).
- What is the number of PCs that gives the minimum RMSE value?
- Make a plot of the regression coefficient paths. An example of this type of plot is in **ISL** figure 6.20(a), page 236. There's also an example in the slides of PCR (slides `18-pcr-regression.pdf`)

4.2) PLSR (20 pts)

- Set a random seed.
- Use `train()` to perform the model building process for PLSR.
- Use `train()`'s argument `tuneLength = 30`.
- Store the `train()` output in an object called `pls_fit`, and print this object.
- Plot the RMSEs against the number of PLS components; use dots connected by a line (see figure 6.11, page 117 for an example of this type of plot).
- What is the number of PLS components that gives the minimum RMSE value?
- Make a plot of the regression coefficient paths. An example of this type of plot is in **ISL** figure 6.20(a), page 236. There's also an example in the slides of PLSR (slides `19-pls-regression1.pdf`)

4.3) Ridge Regression (20 pts)

- Set a random seed.
- Define the candidate set of values for the tuning parameters: see example `ridgeGrid` on page 135.
- Use `train()` to perform the model building process for Ridge Regression.
- Use `ridgeGrid` for the `train()` argument `tuneGrid`.
- Store the `train()` output in an object called `ridge_fit`, and print this object.
- Plot the RMSEs against the values of the tuning parameter `lambda`; use dots connected by a line (see figure 6.16, page 125 for an example of this type of plot)
- What is the value of `lambda` that gives the minimum RMSE value?
- Make a plot of the regression coefficient paths (see figure 6.15, page 124 for an example of this type of plot).

4.4) Lasso (20 pts)

- Set a random seed.
- Define the candidate set of values for the tuning parameters: see example `enetGrid` on page 136.
- Use `train()` to perform the model building process for Lasso.
- Use `enetGrid` for the `train()` argument `tuneGrid`
- Store the `train()` output in an object called `lasso_fit`, and print this object.

- Plot the RMSEs against the values of the tuning parameter `lambda`; use dots connected by a line (see figure 6.16, page 125 for an example of this type of plot)
- What is the value of `lambda` that gives the minimum RMSE value?
- Make a plot of the regression coefficient paths (see figure 6.15, page 124 for an example of this type of plot)

4.5) Model Selection (10 pts)

With the best model of each technique (PCR, PLSR, Ridge, Lasso), use the test data (`solTestXtrans` and `solTestY`) to find which method gives the best overall performance.

- Compute test-MSEs for each method.
- Graph the test-MSEs
- Which method gives the smallest test-MSE?