10 Statistics

10.15 Variable Selection and the Bias-Variance Tradeoff (8 units)

This project requires an understanding of the Part IB Statistics course.

1 Introduction

Consider the following linear model with a univariate response and p covariates:

$$Y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2). \tag{1}$$

We assume throughout that all variables are zero mean. Note also that introducing an intercept is not necessary, since it can be captured by augmenting the covariate and regression vector as follows:

$$b + X\beta = \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} b \\ \beta \end{pmatrix}.$$

We will denote a training dataset of N response-covariate tuples by $\mathcal{T} = \{(y_t, x_t) \mid t = 1, \ldots, N\}$, where each x_t is a $1 \times p$ row vector representing an observation. Alternatively, we may employ matrix notation, letting $\mathbf{y} = (y_t)_{t=1:N}$ be a row vector and $\mathbf{x} = (x_{ti})$ an $N \times p$ matrix where each row corresponds to an observation. The *least squares* (LS) estimate of β then is the minimiser of the *residual sum of squares* (RSS) over the training set:

$$\mathrm{RSS}(\hat{\beta}; \mathcal{T}) = \frac{1}{N}\sum_{t=1}^{N}(y_t - x_t\hat{\beta})^2 = \frac{1}{N}(\mathbf{y} - \mathbf{x}\hat{\beta})^T(\mathbf{y} - \mathbf{x}\hat{\beta}), \text{ and } \hat{\beta}^{\mathrm{LS}}(\mathcal{T}) = \operatorname*{argmin}_{\hat{\beta}} \mathrm{RSS}(\hat{\beta}; \mathcal{T}). \tag{2}$$

Assuming $N > p + 1$, which we do, the LS estimator can be written in closed-form as

$$\hat{\beta}^{\mathrm{LS}}(\mathcal{T}) = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}. \tag{3}$$

The dependence on the training set will be omitted when understood. In this project we will often refer to a subset of covariates as a *model* $\mathcal{M} \subseteq \{1, \ldots, p\}$. The LS estimate for $\mathcal{M}$ is computed on a reduced dataset $\mathcal{T}^{\mathcal{M}}$, obtained by deleting all covariates not in the model:

$$\mathcal{T}^{\mathcal{M}} = \left(y_t, (x_{ti})_{i \in \mathcal{M}}\right)_{t=1:N}.$$

For computational ease, we will instead represent the LS estimate for $\mathcal{M}$ in the p-dimensional space of the original model. We denote this representation by $\hat{\beta}^{\mathcal{M}}$, where

$$\hat{\beta}^{\mathcal{M}}_j(\mathcal{T}) = \begin{cases} \hat{\beta}^{\mathrm{LS}}_{\pi(j)}(\mathcal{T}^{\mathcal{M}}) & \text{if } j \in \mathcal{M}, \\ 0 & \text{otherwise.} \end{cases}$$

Here $\pi : \mathcal{M} \to \{1, 2, \ldots, ||\mathcal{M}||\}$ maps indices of covariates in the model to their respective indices in the reduced dataset $\mathcal{T}^{\mathcal{M}}$, so that $\mathbf{x}\hat{\beta}^{\mathcal{M}}(\mathcal{T})$ is a (simpler) notation for $\mathbf{x}^{\mathcal{M}}\hat{\beta}^{\mathrm{LS}}(\mathcal{T}^{\mathcal{M}})$.

2 The Bias-Variance tradeoff

Question 1 Assume model (1). Let $\hat{\beta}$ be an estimator of β. Show that the expected squared prediction error at a fixed, arbitrary location u can be decomposed as follows:

$$\mathbb{E}_{\mathcal{T},y|u}(y - u\hat{\beta})^2 = \sigma^2 + \left(u\beta - \mathbb{E}_{\mathcal{T}}[u\hat{\beta}]\right)^2 + \mathrm{Var}_{\mathcal{T}}[u\hat{\beta}],$$

where $y \perp \mathcal{T}$. The summands on the right-hand side are often referred to as the *irreducible variance*, *squared estimation bias* and *estimation variance*, respectively. Describe what effect deleting the pth covariate can have on each of these quantities for the LS estimator (i.e., switch $\hat{\beta} = \hat{\beta}^{\mathrm{LS}}$ with $\hat{\beta} = \hat{\beta}^{\{1,\ldots,p-1\}}$). You may consider the simplest non-trivial case, where $\mathbf{x}$ is fixed such that $\mathbf{x}^T\mathbf{x} = \mathbf{I}_p$. It might further be useful to look at $\beta_p = 0$, and then at $\beta_p \neq 0$.

Question 2 Consider model (1) with $p = 10$, $\sigma^2 = 1$, and $X \sim N(0, I_p)$, and set

$$\beta = (-0.5, 0.45, -0.4, 0.35, -0.3, 0.25, -0.2, 0.15, -0.1, 0.05)^T.$$

Simulate a training dataset with $N_{\mathrm{tr}} = 30$ and a test dataset with $N_{\mathrm{te}} = 1000$. Now consider

$$\mathcal{M}_1 = \{1\},\ \mathcal{M}_2 = \{1, 2\},\ \ldots,\ \mathcal{M}_p = \{1, \ldots, p\}.$$

Write a procedure that computes the training and test error of $\hat{\beta}^{\mathcal{M}_j}$ for $j = 1, \ldots, p$. Repeat the experiment 100 times and report your results in a plot of training and test RSS averaged over experiments, against model size. What happens if $N_{\mathrm{tr}} = 200$, and why?

The above demonstrates an effect that holds in much greater generality, namely that suitably reducing the complexity of a model (in this instance, the number of variables involved) can improve prediction accuracy. There may also be gains in *model discovery*, *interpretability*, and, of course, *reduced observation costs*. Consequently, variable selection methods are of interest.

3 Variable selection methods

We consider two approaches to variable selection, *subset selection* and *shrinkage-based methods*. Subset selection methods look among all possible subsets of variables for the one that minimises some suitable estimate of prediction error. The search problem becomes infeasible for large p, and non-exhaustive greedy search methods have to be employed. Shrinkage-based variable selection methods will instead penalise the RSS by a penalty term that forces the LS regression coefficients to shrink in a manner that favours *exact* zeros in $\hat{\beta}$.

3.1 Subset selection

Question 3 **Best subsets selection.** Write a procedure *bestsubset* which takes as input a training dataset $\mathcal{T}$ and outputs a $p \times p$ matrix B, whose jth column contains $\hat{\beta}^{\mathcal{M}_j}$ for the best performing model (in the sense of RSS) of size j, $\mathcal{M}_j$:

$$\mathcal{M}_j(\mathcal{T}) = \underset{\mathcal{M}:\, ||\mathcal{M}||=j}{\mathrm{argmin}}\ RSS\left(\hat{\beta}^{\mathcal{M}}(\mathcal{T}); \mathcal{T}\right).$$

What is the size of the model space $\{\mathcal{M} \mid \mathcal{M} \subseteq \{1, \ldots, p\}\}$? Your procedure will handle with difficulty values of p for which the size of the search space $\{\mathcal{M} \mid \mathcal{M} \subseteq \{1, \ldots, p\}, ||\mathcal{M}|| = j\}$ exceeds 10^5 for any $j \in \{1, \ldots, p\}$. What is the smallest such p (show your work)?

Question 4 **Greedy subset selection**. Write a procedure *greedysubset*, using the same input-output format as before, that incrementally builds up the model sequence $\mathcal{M}_j$ by adding at each iteration the covariate that improves model fit the most:

$$\mathcal{M}_0 = \emptyset, \quad \mathcal{M}_{d+1}(\mathcal{T}) = \mathcal{M}_d(\mathcal{T}) \cup \left\{ l \mid l = \operatorname*{argmin}_j RSS\left(\hat{\beta}^{\mathcal{M}_d(\mathcal{T})\cup\{j\}}(\mathcal{T}); \mathcal{T}\right) \right\}.$$

Can the fact that the family of models $\mathcal{M}_0, \ldots, \mathcal{M}_p$ is nested be used to gain in computational efficiency? Explain how, without effecting the change. Assuming that $\mathcal{M}_j = \{1, \ldots, j\}$, you might want to consider the upper left $j \times j$ block of $\left((x^{\mathcal{M}_{j+1}})^T x^{\mathcal{M}_{j+1}}\right)^{-1}$.

Question 5 **Forward F-test.** Amend *greedysubset* to stop whenever the newly added variable does not significantly improve fit (at the $p = .05$ level), using the F-statistic

$$\frac{\mathrm{RSS}(\hat{\beta}^{\mathcal{M}_d}) - \mathrm{RSS}(\hat{\beta}^{\mathcal{M}_{d+1}})}{\mathrm{RSS}(\hat{\beta}^{\mathcal{M}_{d+1}})/(N-d-1)},$$

which you may assume follows an $F_{1,N-d-1}$ distribution (you may use the MATLAB function *cdf*). Would this method work for best subset selection?

Question 6 We can represent a sparse (linear regression) estimator more generally as an algorithm that takes as input a training set $\mathcal{T}$ and outputs a sequence of p candidate regression vectors for each model size (i.e., the jth candidate $\hat{\beta}^{(j)}(\mathcal{T})$ has precisely $p - j$ zeros). Best and greedy subset search are special cases of this definition for which each candidate is a least squares solution, a condition we will not insist on here. We would like to select among candidates on the basis of estimated prediction error $\hat{\mathrm{PE}}$:

$$\hat{\beta}^{\mathrm{CV}}(\mathcal{T}) = \hat{\beta}^{j^\star}(\mathcal{T}), \text{ where } j^\star = \operatorname*{argmin}_j \left\{ \hat{\mathrm{PE}}(j, \mathcal{T}) \right\}.$$

The prediction error can be estimated using 10-fold cross-validation as

$$\hat{\mathrm{PE}}(j, \mathcal{T}) = \frac{1}{10} \sum_{k=1}^{10} RSS\left(\hat{\beta}^{(j)}(\mathcal{T}^{-k}); \mathcal{T}^k\right),$$

where T^k is the kth fold of the training set and $\mathcal{T}^{-k}$ its complement:

$$\begin{aligned} \mathcal{T}^k &= \left\{ (y_{\pi(n)}, x_{\pi(n)}) \mid k - 1 < \frac{10n}{N} \leqslant k \right\}, \\ \mathcal{T}^{-k} &= \left\{ (y_{\pi(n)}, x_{\pi(n)}) \mid \frac{10n}{N} \leqslant k - 1 \text{ or } \frac{10n}{N} > k \right\}, \end{aligned} \tag{4}$$

where π is a random permutation of $\{1, \ldots, N\}$ (you may use the MATLAB function *randperm*). Implementing the above for an arbitrary sparse estimator would involve a function taking another function as an argument. In MATLAB, this can be achieved using *function handles*, as demonstrated by *handle_demo.m* and *testerror.m* available from the CATAM website. Write a procedure *crossval* that implements the above (the MATLAB functions *ismember* and *find* might be useful in this). This procedure should take as input $\mathcal{T}$ and a sparse estimator, and output $\hat{\beta}^{\mathrm{CV}}(\mathcal{T})$.

3.2 The Lasso estimator

The Lasso estimator penalises the RSS by the $L1$ norm of the regression coefficients:

$$\hat{\beta}^{(\mathrm{L},\lambda)}(\mathcal{T}) = \underset{\hat{\beta}}{\operatorname{argmin}} \left\{ \mathrm{RSS}\left(\hat{\beta}; \mathcal{T}\right) + \lambda \sum_{j=1}^{p} \left|\hat{\beta}_j\right| \right\} \tag{5}$$

Question 7 Express the Lasso as a quadratic program with linear constraints.

In the Lasso estimator, the degree of sparsity is controlled *indirectly* via the penalty weight λ, rather than directly as in earlier methods. For $\lambda = 0$ the full model is employed, whereas increasingly many covariates are deleted from the model as $\lambda \to \infty$. Given an algorithm for solving (5), we can then use cross-validation to select among any finite set of values $\lambda_1 < \lambda_2 < \cdots < \lambda_q$ for λ. For simplicity, we will continue here to perform cross-validation to select model size rather than λ. To do so, we will rely on the LARS algorithm, which, subject to certain minor assumptions and modifications that do not concern us here, allows us to compute in an efficient manner one Lasso solution for each model size. The file *monotonic_lars.m* available from the CATAM website contains an implementation of this modified LARS algorithm that can be used as input to *crossval*.

Question 8 The file *prostate.dat* available from the CATAM website contains a prostate cancer dataset.* The dataset has been preprocessed to standardise the covariates and make all variables zero mean, so that you can avoid using an intercept. Column 1 contains the response, *lpsa*, and columns 2 to 9 the covariates *lcavol*, *lweight*, *age*, *lbph*, *svi*, *lcp*, *gleason*, and *pgg45*. Augment the dataset by adding four zero-mean, unit-variance covariates sampled from a distribution of your liking *independently of variables in prostate.dat*. Separate the data into a training dataset of size 70 and a test dataset of size 27. Perform a variable selection analysis of the data using the tools developed above. Present and discuss your results.

*reproduced from `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`

