Notes of the Elements of Statistical Learning

Tianyu Wang

January 9, 2020

Contents

Chapter 1

Overview of Supervised Learning

1.1 Least Squares and Nearest Neighbors

Consider two scenario:

1. **(Better on linear regression)** The training data was generated from bivariate Gaussian distributions with uncorrelated components and different means.
2. **(Better on nearest-neighbor)** The training data came from a mixture of multiple low-variance Gaussian distributions, with individual means themselves distributed as a Gaussian.

1.1.1 Linear Regression Models and Least Squares

Given $X = (X_1, X_2, ..., X_P)$, predict the output Y via model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j = X^T \hat{\beta}$$

- When $\beta_0 = 0$, hyperplane $(X, \hat{Y})$ forms a subspace. Otherwise, it is an affine set.

Least squares: $RSS(\beta) = (Y - X\beta)^T(Y - X\beta) = ||Y - X\beta||_2^2$

- Quatratic⇒Minimum always exists.

Unique solution: $\hat{\beta} = (X^T X)^{-1} X^T Y$

- Calculated by differentiating $RSS(\beta)$ with respect to β.
- Minimizing $RSS(\beta)$ ⇔ choosing $\hat{\beta}$ so that $Y - \hat{Y}$ is orthogonal to the subspace spanned by column vectors of X ⇔ $\dfrac{\partial RSS}{\partial \beta} = -2X^T(Y - X\beta) = 0$.
- **Do not need a large data set to fit**.

1.1.2 Nearest-Neighbor Methods

Use observations in the training set $\mathcal{T}$ closest to x to form $\hat{Y}$.

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- Not convex
- Effective number of parameters of kNN is N/k (explanation).
- Error on train data should be an increasing function of k, and equals 0 for $k = 1$.
- Unnecessarily noisy for Gaussian data.

1.1.3 From Least Squares to Nearest Neighbors

Linear boundary from least squares: smooth, relies heavily on the linear assumption. Low variance and high bias.

kNN: Does not rely on assumptions about underlying data. Any subregion depends on input points and their position. High variance and low bias.

- 1-nearest-neighbor captures a large percentage of the market for low-dimensional problems.

Ways in which these procedures are enhanced

- Kernel methods use weights that decrease smoothly to zero with distance from the target method instead of 0/1 in kNN.
- In high dimensional spaces the distance kernels are modified to emphasize some variables more than others.
- Local regression fits linear models by locally weighted least squares rather than fitting constants locally.
- Basis expansion of the inputs to improve complexity of linear model.
- Projection pursuit/NN models: nonlinear transform of linear models

1.2 Statistical Decision Theory

1.2.1 Quantitative output

Let $X \in \mathbb{R}^p, Y \in \mathbb{R}$ with joint distribution $Pr(X, Y)$.
Goal: Find a function $f(X)$ for predicting Y, which requires a *loss function* $L(Y, f(X))$.
Squared error loss:

$$\begin{aligned} L(Y, f(X)) =& (Y - f(X))^2 \\ EPE(f) =& E(Y - f(X))^2 = \int [y - f(x)]^2 Pr(dx, dy) \\ =& \int [y - f(x)]^2 Pr(dy|dx) Pr(dx) = E_X E_{Y|X}([Y - f(X)]^2|X) \end{aligned}$$

where EPE stands for Expected Prediction Error. I$u(x, t) = 0$ when $x < 0$ t suffices to minimize EPE pointwise

Thus when the measure is squared error, $E(Y|X = x)$ is the best solution.

kNN:

$$\hat{f}(x) = \text{Avg}(y_i|x_i \in N_k(x))$$

- Expectation is approximated by averaging
- Conditioning at a point is relaxed to a region
- As $N, k \to \infty$ such that $k/N \to 0$, $\hat{f}(x) \to E(Y|X = x)$
  - Since we do not have many samples in most cases, we can usually get a more stable model if some more structured model is appropriate.
- Assumes $f(x)$ is well approximated by a locally constant function.

Linear Regression:Let $f(x) = x^T\beta$ and differentiating $E(Y - f(X))^2$, we have optimal β

$$\beta = [E(XX^T)]^{-1}E(XY)$$

The least squares solution just replace the expectation by averages on the training data.

- Assumes $f(x)$ is well approximated by a globally linear function

If we use L_1 loss instead of L_2, the solution will becomes the conditional median

$$\hat{f}(x) = \text{median}(Y|X = x)$$

- A different measure of location
- More robust than conditional mean

1.2.2 Categorical output

Let the categorical variable be G, $K = card(\mathcal{G})$, $K \times K$ matrix L be its loss function, $L_{ii} = 0, L_{ij} \geq 0$, $L(k, l)$ be the cost of wrong prediction(0/1 in most cases).

$$\begin{aligned} EPE =& E[L(G, \hat{G}(X))] \\ =& E_X \sum_{k=1}^{K} L[\mathcal{G}_k, \hat{G}(X)] Pr(\mathcal{G}_k|X) \\ \hat{G}(x) =& \arg\min_{g \in \mathcal{G}} \sum_{k=1}^{K} L[\mathcal{G}_k, g] Pr(\mathcal{G}_k|X = x) \end{aligned}$$

With 0-1 loss, it can by simplified as

$$\begin{aligned} \hat{G}(x) =& \arg\min_{g \in \mathcal{G}} [1 - Pr(g|X = x)] \\ =& \mathcal{G}_k \text{ if } Pr(\mathcal{G}_k|X = x) = \max_{g \in \mathcal{G}} Pr(g|X = x) \end{aligned}$$

It is known as *Bayes classifier*, says we classify the most probable class with conditional distribution $Pr(G|X)$. The error rate here is called *Bayes rate*.

1.3 Local Methods in High Dimensions

In **low dimension space**, kNN can always find the theoretically optimal conditional expectaion with a large training set.

- Not work in high dimentional data: *Curse of dimensionality*

Ex 1.3.1. Consider N data points uniformly distributed on a unit ball, the median distance from the origin to the closet data point is

$$d(p,N) = (1 - \frac{1}{2}^{1/N})^{1/p}$$

- Let $N = 500, p = 10 \Rightarrow d(p,N) = 0.52$ ⇒Most points are closer to boundary but not other points⇒Must extrapolate from samples but not interpolate between them.
- Sampling density is proportional to $N^{1/p}$

Ex 1.3.2. Consider 1000 x_i from $U([-1,1]^p)$, $Y = f(X) = exp(-8||X||^2)$, then use 1-NN to predict y_0 at $x_0 = 0$, we have *bias-variance decomposition*

$$\begin{aligned} MSE(x_0) =& E[f(x_0) - \hat{y}_0]^2 = E[\hat{y}_0 - E[\hat{y}_0]]^2 + [E[\hat{y}_0] - f(x_0)]^2 \\ =& Var(\hat{y}_0) + Bias^2(\hat{y}_0) \end{aligned}$$

When dimension increase, we will find that distance of the closet point to 0 increase. The bias will also significantly increase.

- Complexity of functions of many variables grow exponentially with the dimension–Need exponential growth of the size of the training set to maintain accuracy.

If we use function which only cares about a few dimensions like $f(X) = \frac{1}{2}(X_1 + 1)^3$, the bias will be stable, but the variance will increase a lot.

Linear model

Let $Y = X^T\beta + \epsilon, \epsilon \sim N(0,\sigma^2)$, than for test point x_0, $\hat{y}_0 = x_0^T\hat{\beta}_0 = x_0^T\beta + \sum_{i=1}^N l_i(x_i)\epsilon_i$, l_i is the ith element of $X(X^TX)^{-1}x_0$. Since the model under the least squares is unbiased, we have

$$\begin{aligned} EPE(x_0) =& E_{y_0|x_0}E_{\mathcal{T}}(y_0 - \hat{y}_0)^2 \\ =& Var(y_0|x_0) + E_{\mathcal{T}}[\hat{y}_0 - E_{\mathcal{T}}\hat{y}_0]^2 + [E_{\mathcal{T}}\hat{y}_0 - x_0^T\beta]^2 \\ =& Var(y_0|x_0) + Var_{\mathcal{T}}(\hat{y}_0) + Bias^2(\hat{y}_0) \\ =& \sigma^2 + E_{\mathcal{T}}x_0^T(X^TX)^{-1}x_0\sigma^2 + 0^2 \end{aligned}$$

When N is large and $\mathcal{T}$ is selected at random, assume $E[X] = 0$, then $X^TX \to NCov(X)$

$$\begin{aligned} E_{x_0}EPE(x_0) \sim& E_{x_0}x_0^TCov(X)^{-1}x_0\sigma^2/N + \sigma^2 \\ =& trace[Cov(X)^{-1}Cov(x_0)]\sigma^2/N + \sigma^2 \\ =& \sigma^2(p/N) \end{aligned}$$

Thus expected EPE increases as a linear function of p with slope σ^2/N.

1.4 Statistical Models, Supervised Learning and Function Approximation

Goal: find a $\hat{f}(x)$ of $f(x)$. Squared error lead to $f(x) = E(Y|X = x)$, but they may fail

- If the dimension is too high, nearest neighbors can not be close to the target and cause large error.
- If the structure is known, this can be used to reduce bias and variance.

1.4.1 A Statistical Model for the Joint Distribution $Pr(X, Y)$

Suppose our data comes from $Y = f(X) + \epsilon$

Chapter 2

Linear Methods for Regression

Useful in situations with small numbers of training cases, low signal-to-noise ratio, or sparse data.

2.1 Linear Regression Models and Least Squares

Consider linear model

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

which assume the regression function $E(Y|X)$ linear or the linear model is a good approximation. Here X_j can be

- Quantitative inputs
- Transformation/Basis expansion of inputs e.g. log, square-root, polynomial
- Numeric/dummy coding of the levels of the inputs

$$RSS(\beta) = (y - X\beta)^T (y - X\beta)$$

X has full column rank$\Rightarrow X^T X$ positive definite.

By solution, we have $\hat{y} = X\beta = X(X^T X)^{-1} X^T Y, \hat{y}_i = x_i^T \hat{\beta}$

- $H = X(X^T X)^{-1} X^T$: hat matrix, which computes the orthogonal projection from Y to $\hat{Y}$.

When columns of X are not independent

- Perfectly correlated$\Rightarrow X^T X$ singular, $\hat{\beta}$ not uniquely defined. However, $\hat{y}$ are still the projection from y to column space of X.
  - May appear when number of inputs p exceed the number of training cases N, where features reduced by filtering or fitting controlled by regularization.

If y_i are uncorrelated with constant variance σ^2 and x_i are fixed (non random), then we have estimation

$$Var(\hat{\beta}) = E[(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T] = E[((X^T X)^{-1} X^T \epsilon)((X^T X)^{-1} X^T \epsilon)^T] = (X^T X)^{-1} \sigma^2$$

$$\hat{\sigma}^2 = \frac{1}{N - p - 1} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

where $N-p-1$ makes $\hat{\sigma^2}$ is an unbiased estimation because $\sum_{i=1}^{N}(y_i-\hat{y}_i)^2 \sim \sigma^2\chi^2_{N-p-1}$.

Now we assume the linearity of the model and deviations of Y around its expectation are additive and Gaussian, consider the model

$$Y = E(Y|X_1,...,X_p) + \epsilon = \beta_0 + \sum_{j=1}^{p} X_j\beta_j + \epsilon$$

where $\epsilon \sim N(0,\sigma^2)$, then $\hat{\beta} \sim N(\beta,(X^TX)^{-1}\sigma^2)$, $(N-p-1)\hat{\sigma}^2 \sim \sigma^2\chi^2_{N-p-1}$, and $\hat{\beta},\hat{\sigma}^2$ statistically independent.

To test **whether** $\beta_j = 0$, we consider Z-score

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}}$$

where v_j is the jth diagonal element of $(X^TX)^{-1}$, under null hypothesis, z_j is distributed as t_{N-p-1}, thus a large value of z_j will reject the null hypothesis.

- When $\hat{\sigma}$ is replaced by σ, z_j is normally distributed.
- Difference of tail quantiles will be small when sample size increases.

To test the **significance of groups of coefficients**, e.g. whether a variable can be excluded, we need to test whether its coefficients can all be set to zero, here we use F-statistics

$$F = \frac{(RSS_0 - RSS_1)/(p_1-p_0)}{RSS_1/(N-p_1-1)}$$

where RSS_1 is the residual sum of squares with p_1+1 parameters, RSS_0 for the nested smaller model with p_0+1 parameters, having p_1-p_0 constrained to be 0, F statistics measures the change in residual sum of squares per additional parameter in the bigger model.

Under Gaussian assumption and null hypothesis that smaller model is correct, $F \sim F_{p_1-p_0,N-p_1-1}$.

- z_j is equivalent to F statistics when only drop one coefficient.
- When N is large enough, quantiles of $F_{p_1-p_0,N-p_1-1}$ approach those of the $\chi^2_{p_1-p_0}$

When we can isolate β_j to obtain a $1-2\alpha$ confidence interval for β_j

$$(\hat{\beta}_j - z^{(1-\alpha)}v_j^{\frac{1}{2}}\hat{\sigma}, \hat{\beta}_j + z^{(1-\alpha)}v_j^{\frac{1}{2}}\hat{\sigma})$$

where $z^{(1-\alpha)}$ is the $1-\alpha$ percentile of normal distribution.

We can also obtain an approximate confidence set for the entire parameter vector β

$$C_\beta = \{\beta|(\hat{\beta}-\beta)^T(\hat{\beta}-\beta) \leq \hat{\sigma}^2{\chi^2_{p+1}}^{(1-\alpha)}\}$$

This confidence set for β generates a corresponding confidence set for $f(x) = x^T\beta$, namely $\{x^T\beta|\beta \in C_\beta\}$

Another way of comparing significance between different variables: Use Z-score=mean/std of coefficient and compare Z-score of different variables. Z-score>2 implies significance at 5% level

2.1.1 The Gauss-Markov Theorem

- Least squares estimates of β have the smallest variance among all linear unbiased estimates.
- Unbiased estimation may not be a wise choice.

Consider linear combination of parameters $\theta = a^T\beta$, then the least square estimates of $a^T\beta$ is

$$\begin{aligned}\hat{\theta} =& a^T\hat{\beta} = a^T(X^TX)^{-1}X^TY \\ E(a^T\hat{\beta}) =& E(a^T(X^TX)^{-1}X^Ty) = a^T(X^TX)^{-1}X^TE(y) = a^T(X^TX)^{-1}X^TX\beta = a^T\beta\end{aligned}$$

Thus $a^T\hat{\beta}$ is unbiased if the linear model is correct.

Thm 2.1.1. *(Gauss-Markov theorem) If we have any other unbiased linear estimator $\tilde{\theta} = c^Ty$ for $a^T\beta$, then $Var(a^T\hat{\beta}) \leq Var(c^Ty)$*

Proof. Triangle inequality. □

Rmk. Can be extended to entire parameter β with a few more definition.

Now consider

$$MSE(\tilde{\theta}) = E(\tilde{\theta} - \theta)^2 = Var(\tilde{\theta}) + [E(\tilde{\theta}) - \theta]^2$$

There may exist estimators with little bias and huge reduction on variance, which brings them less MSE. Recall that for $Y_0 = f(x_0) + \epsilon_0, \tilde{f}(x_0) = x_0\tilde{\beta}$,

$$E(Y_0 - \tilde{f}(x_0))^2 = \sigma^2 + E(x_0^T\tilde{\beta} - f(x_0))^2 = \sigma^2 + MSE(\tilde{f}(x_0))$$

We have prediction error is related to MSE.

2.1.2 Multiple Regression from Simple Univariate Regression

Consider univariate model $Y = X\beta + \epsilon$. Then we have

$$\hat{\beta} = \frac{< x, y >}{< x, x >}, \quad r = t - x\hat{\beta}$$

We can generate a similar model for a p-variable model.

By Schmidt orthogonalization, we can generate an orthogonal basis for the column space, and

$$\hat{\beta}_p = \frac{< z_p, y >}{< z_p, z_p >}$$

It is actually the regression coefficient of y on x_p. We can also see that jth coefficient is the residual after regressing x_j on $x_0, x_1, ..., x_{j-1}, x_{j+1}, ...x_p$.

If x_p is highly correlated to some other x_k's, residual vector z_p should be close to 0, and $\hat{\beta}_p$ would be very unstable.

$$Var(\hat{\beta}_p) = \frac{\sigma^2}{< z_p, z_p >}$$

It means the precision with which we can estimates $\hat{\beta}_p$ depends on how much of x_p can be explained by other x_k's.

We may also apply QR decomposition on $X = QR$, where Q is an $N \times (p+1)$ orthogonal matrix, $Q^TQ = I$, R is a $(p+1) \times (p+1)$ upper triangular matrix. Then the solution is given by

$$\hat{\beta} = R^{-1}Q^Ty, \quad \hat{y} = QQ^Ty$$

2.1.3 Multiple Outputs

Suppose we want to predict $Y_1, ..., Y_k$ with $X_0, X_1, ...X_p$, we assume a linear model

$$\begin{aligned} Y_k =& \beta_{0k} + \sum_{j=1}^{p} X_j \beta_{jk} + \epsilon_k = f_k(X) + \epsilon_k \\ Y =& XB + E \\ RSS(B) =& tr[(Y - XB)^T (Y - XB)] \\ \hat{B} =& (X^T X)^{-1} X^T Y \end{aligned}$$

Hence the coefficients for Y_k does not depend on other variable's least square estimation.

If $Cov(\epsilon) = \Sigma$, we have

$$RSS(B; \Sigma) = \sum_{i=1}^{N} (y_i - f(x_i))^T \Sigma^{-1} (y_i - f(x_i))$$

It arises naturally from Gaussian. If Σ_i vary from observation, then $\hat{B} = (X^T X)^{-1} X^T Y$ is no longer the case.

2.2 Subset Selection

Why we are always not satisfied with OLS estimation

- *Low bias but large variance.* Can be improved by shrinking/setting some coefficients to zero, which sacrifice a little bit of bias but reduce the variance.
- *Interpretion.* We want to get a smaller subset that exhibit the strongest effects from many predictors. AKA sacrifice small details to get the "big picture".

Subset selection is a discrete process and may still have high variance, so doesn't reduce the prediction error of the full model.

2.2.1 Best-Subset Selection

Find each $k \in \{0, 1, 2, ...p\}$, the subset of size k gives smallest RSS. Note that RSS is always a decreasing function of k, which makes it cannot be used to choose k. Choosing k is about the trade-off between bias and variance. Typically we use the smallest model that minimizes an estimate of the expected prediction error.

2.2.2 Forward- and Backward-Stepwise Selection

Here we try to find a good path through subsets but not search all of them.

Forward-stepwise selection: Starts with the intercept, then adds the predictor that most improves the fit. Methods like QR decomposition an make this fast.

- Can always be used
- It is a greedy approach, which might be sub-optimal.
- However, it is computational, and has lower covariance(perhaps higher bias).

Backward-stepwise selection: Starts with the whole model, then deletes the predictor that least impacts the fit(variable with lowest Z-score).

- Only used when $N > p$.
- Similar performance

Some packages can do both at the same time. Criterion like AIC could be a good idea, which takes proper account of the number of parameters. Add or drop will be performed that minimizes the AIC. On the other hand, F-statistics is kind of out of fashion.

Forward-Stagewise Regression

Slow and inefficient, but may pay dividends in high-dimensional problems.

2.3 Shrinkage Methods

More continuous compared with subset selection, doesn't suffer as much from high variability.

2.3.1 Ridge Regression

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \left\{ ||y - X\beta||_2^2 + \lambda ||\beta||_2^2 \right\}$$

Idea of penalizing parameters is also used in neural network as *weight decay*.

The problem is equivalent to

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \left\{ ||y - X\beta||_2^2 \right\} \text{ subject to } ||\beta||_2^2 < t$$

where t and λ have one-to-one correspondence. When there are correlated variables, there may be mildly large and small parameters, Imposing a size constraint can solve this problem.

The solution is

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

λI makes the problem always nonsingular even if $X^T X$ is singular.

In the case of orthonormal inputs, Ridge is just a scaled OLS estimates $\hat{\beta}^{\text{ridge}} = \hat{\beta}/(1+\lambda)$.

SVD insight Consider SVD $X = UDV^T$, here U span the column space of X, V span the row space, $d_1 \geq d_2 \geq ... \geq d_p \geq 0$. When there is $d_j = 0$, X is singular. U and Q(in QR decomposition) are generally different orthogonal bases for the column space of X.

With SVD, we have $X\hat{\beta}^{\text{ls}} = UU^T y$, and ridge solution

$$X\hat{\beta}^{\text{ridge}} = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T y$$

It shows Ridge shrinks the coordinates by $d_j^2/(d_j^2 + \lambda)$. It shows greater amount is shrinked for small d_j^2. Note that $X^T = VD^2V^T$. v_j the PC directions of X. $z_1 = Xv_1 = u_1 d_1$ explains the most variance, and $Var(z_1) = d_1^2/N$. z_1 the first PC of X.

$\sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$: *Effective degrees of freedom*

2.3.2 Lasso Regression

$$\begin{aligned}\hat{\beta}^{\text{lasso}} &= \arg\min_{\beta}\left\{||y - X\beta||_2^2\right\} \text{ subject to } ||\beta||_1 < t \\ &= \arg\min_{\beta}\left\{||y - X\beta||_2^2 + \lambda||\beta||_1\right\}\end{aligned}$$

Does a kind of continuous subset selection. $\hat{\beta}_j^{\text{lasso}} = sign(\hat{\beta}_j)(\left|\hat{\beta}_j\right| - \lambda)_+$

2.3.3 Discussion: Subset Selection, Ridge Regression and the Lasso

Actually, $\hat{\beta}_j^{\text{bestSubset}} = \hat{\beta}_j \cdot I(\left|\hat{\beta}_j\right| \geq \hat{\beta}_{(M)})$

$$\hat{\beta}^{\text{bestSubset}} == \arg\min_{\beta}\left\{||y - X\beta||_2^2 + \lambda\sum_j \left|\beta_j\right|^0\right\}$$

$\lambda\sum_j\left|\beta_j\right|^0$: counts of nonzero values.

Elastic-penalty:

$$\lambda\sum_{j=1}^{p}(\alpha\beta_j^2 + (1-\alpha)\left|\beta_j\right|)$$

It is a compromise between ridge and lasso: selects variables like lasso, and shrinks together the coefficient of correlated predictors like ridge. Has computational advantages over L_q penalties.

We may not consider $q \in [0, 1)$, in which case the function is not convex, which will cause some problems during optimization.

2.3.4 Least Angle Regression(LAR)

A "democratic" version of forward stepwise regression. Extremely efficient.

Algorithm 3.2 *Least Angle Regression.*

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.
2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.
3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r}\rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.
4. Move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.
5. Continue in this way until all p predictors have been entered. After $\min(N-1, p)$ steps, we arrive at the full least-squares solution.

Let $\mathcal{A}_k$ be the active set at step k, $\beta_{\mathcal{A}_k}$ be the coefficient vector. There will be $k-1$ nonzero values and one zero. Let $r_k = y - X_{\mathcal{A}_k}\beta_{\mathcal{A}_k}$, then the direction for this step is

$$\delta_k = (X_{\mathcal{A}_k}^T X_{\mathcal{A}_k})X_{\mathcal{A}_k}^T r_k$$

Then coefficient profile evolves as $\beta_{\mathcal{A}_k}(\alpha) = \beta_{\mathcal{A}_k} + \alpha \cdot \delta_k$. Let $u_k = X_{\mathcal{A}_k}\delta_k$, it makes the smallest and equal angle with each predictor in $\mathcal{A}_k$. With LAR, we can know the step length at the beginning of each step.

2.4 Methods of Derived Input Directions

Large number of inputs with high correlation. This section describes how to linearly combine X_j to Z_m used in regression.

2.4.1 Principal Components Regression

Since Z_m here are orthogonal, when regress y for some $M < p$,

$$\hat{y}_{(M)}^{\text{per}} = \bar{y}1 + \sum_{m=1}^{M} \hat{\theta}_m z_m, \quad \hat{\theta}_m = \frac{< z_m, y >}{< z_m, z_m >}$$

Since $z_m = Xv_m$, we have

$$\hat{\beta}^{\text{per}}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m$$

Compared with Ridge, it discards the $p - M$ smallest eigenvalue components.

2.4.2 Partial Least Squares

Assume x_j is normalized(PLS and principal components regression are not scale invariant). It begins by

$$\hat{\varphi}_{1j} =< x_j, y >, \quad z_1 = \sum_j \hat{\varphi}_{1j} x_j$$

Hence in the construction, inputs are weighted by the strength of their univariate effect on y.

Algorithm 3.3 *Partial Least Squares.*

1. Standardize each $\mathbf{x}_j$ to have mean zero and variance one. Set $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$, and $\mathbf{x}_j^{(0)} = \mathbf{x}_j,\ j = 1, \ldots, p$.
2. For $m = 1, 2, \ldots, p$
   (a) $\mathbf{z}_m = \sum_{j=1}^{p} \hat{\varphi}_{mj}\mathbf{x}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y}\rangle$.
   (b) $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y}\rangle / \langle \mathbf{z}_m, \mathbf{z}_m\rangle$.
   (c) $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m\mathbf{z}_m$.
   (d) Orthogonalize each $\mathbf{x}_j^{(m-1)}$ with respect to $\mathbf{z}_m$: $\mathbf{x}_j^{(m)} = \mathbf{x}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{x}_j^{(m-1)}\rangle / \langle \mathbf{z}_m, \mathbf{z}_m\rangle]\mathbf{z}_m,\ j = 1, 2, \ldots, p$.
3. Output the sequence of fitted vectors $\{\hat{\mathbf{y}}^{(m)}\}_1^p$. Since the $\{\mathbf{z}_\ell\}_1^m$ are linear in the original $\mathbf{x}_j$, so is $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\hat{\beta}^{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.

PLS seeks directions that have high variance and high correlation with the response. In particular, the mth principal component direction v_m solves

$$\max_{\alpha} Var(X\alpha) \quad \text{subject to } ||\alpha|| = 1,\ \alpha^T S v_l = 0,\ l = 1, 2, ..., m-1$$

where S is the covariance matrix of x_j. $\alpha^T S v_l$ shows that $z_m = X_\alpha$ is uncorrelated with all $z_l = Xv_l$.

On the other hand, the mth PLS direction solves

$$\max_{\alpha} Corr^2(y, X\alpha)Var(X\alpha) \quad \text{subject to } ||\alpha|| = 1,\ \alpha^T S v_l = 0,\ l = 1, 2, ..., m-1$$

Further study shows that the variance part tends to dominant, so PLS behaves like Ridge/PCR.

2.5 A Comparison of the Selection and Shrinkage Methods

Ridge shrinks all directions, but shrinks low-variance directions more.

PCR leaves M high-variance directions alone, and discards the rest.

PLS also tends to shrink the low-variance directions, but can actually inflate some of the higher variance directions. This can make it a little unstable, and cause it to have slightly higher prediction error compared to ridge regression.

Conclusion: PLS, PCR and ridge regression tend to behave similarly. Ridge regression may be preferred because it shrinks smoothly, rather than in discrete steps. Lasso falls somewhere between ridge regression and best subset regression, and enjoys some of the properties of each.

2.6 Multiple Outcome Shrinkage and Selection

Simple Approach: Apply a univariate technique individually to each outcome or simultaneously to all outcomes.

Or we can exploit correlations in the different responses.

2.7 More on the Lasso and Related Path Algorithms

2.8 Computational Considerations

Least squares fitting is usually done via Cholesky decomposition of $X^T X(p^e + Np^2/2)$ or QR decomposition(Np^2). Lasso with LAR has the same order of computation.

Chapter 3

Linear Methods for Classification

3.1 Introduction

Decision boundaries are linear in this chapter.

Suppose there are K classes in discrete set $\mathcal{G}$, and the fitted linear model is $\hat{f}_k(x) = \hat{\beta}_{k0} + \hat{\beta}_k^T x$. The decision boundary for class k and l should then be the set where $\hat{f}_k(x) = \hat{f}_l(x)$, which is an affine set or hyperplane. This approach is a member of a class of methods that model *discriminant functions* $\delta_k(x)$ for each class. Methods that model $Pr(G = k|X = x)$ are also in this clear. All we require is some monotone transformation of δ_k or $Pr(G = k|X = x)$ be linear for the boundaries to be linear. For example, we consider *logit transformation* $log[p/(1-p)]$, we can see that

$$\begin{aligned} Pr(G = 1|X = x) =& \frac{exp(\beta_0 + \beta^T x)}{1 + exp(\beta_0 + \beta^T x)} \\ Pr(G = 2|X = x) =& \frac{1}{1 + exp(\beta_0 + \beta^T x)} \\ log\frac{Pr(G = 1|X = x)}{Pr(G = 2|X = x)} =& \beta_0 + \beta^T x \end{aligned}$$

3.2 Linear Regression of an Indication Matrix

$Y = (Y_1, ..., Y_k)$, $Y_k = 1(G = k)$. Then

$$\hat{Y} = X(X^T X)^{-1} X^T Y$$

Then for a new observation x,

$$\hat{f}(x)^T = (1, x^T)\hat{B}, \quad \hat{G}(x) = argmax_{k \in \mathcal{G}} \hat{f}_k(x)$$

We can verify that $\sum_{k \in \mathcal{G}} \hat{f}_k(x) = 1$, however, in real world, $\hat{f}_k(x) = 1$ can be larger than 1 or negative. We may solve the problem by basis expansions of the inputs.

A more simple way is to construct targets t_k for each class(t_k the kth row of I_k). Then we try to fit

$$\min_B \sum_{i=1}^N ||y_i - [(1, x_i^T)B]^T||^2, \quad \hat{G}(x) = argmin_k ||\hat{f}(x) - t_k||^2$$

When K becomes large, classes can be masked by others because of the regression model. Polynomials up to degree $K - 1$ might be needed tro solve them. The worst complexity could be $O(p^{K-1})$.

3.3 Linear Discriminant Analysis

We need to know $Pr(G|X)$ for optimal classification. Suppose $f_k(x)$ is the density of X in class $G = k$, π_k be the prior probability of class k, $\sum \pi_k = 1$. Then by Bayes

$$Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

Many techniques are based on models for the class densities.

- Linear/Quadratic discriminant analysis: Gaussian
- Nonlinear decision boundaries: Mixture of Gaussians
- General nonparametric density estimates for each class allow the most flexibility.
- Naive Bayes assumes inputs are conditionally independent in each class.

Suppose each class density as $N(\mu_j, \Sigma_k)$. LDA arises when $\Sigma_k = \Sigma$. Then we can see that

$$\begin{aligned} log\frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} &= \log \frac{f_k(x)}{f_l(x)} + \log(\pi_k)\pi_l \\ &= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T\Sigma^{-1}(\mu_k - \mu_l) + x^T\Sigma^{-1}(\mu_k - \mu_l) \end{aligned}$$

which is an linear function of x. Therefore the decision boundaries are also linear. They would be the perpendicular bisectors of the line segments joining the centroids if $\Sigma = \sigma^2 I$. We see the linear discriminant functions

$$\delta_k = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log \pi_k$$

In practice we estimate the parameters by

$$\begin{aligned} \hat{\pi}_k &= \frac{N_k}{N} \\ \hat{\mu}_k &= \sum_{g_i=k} x_i/N_k \\ \hat{\Sigma} &= \sum_k \sum_{g_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T/(N - K) \end{aligned}$$

With more than two classes, LDA is not the same as linear regression of the class indicator matrix, and it avoids the masking problems associated with that approach.

When Σ_k are not similar, we get *quadratic discriminant functions(QDA)*

$$\delta_k(x) = -\frac{1}{2}|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T\Sigma_k^{-1}(x - \mu_k) + \log \pi_k$$

Then decision boundary is determines by a quadratic equation.

Both LDA and QDA perform well on an amazingly large and diverse set of classification tasks. A reason is that the data can only support simple decision boundaries such as linear or quadratic, and the estimates provided via the Gaussian models are stable. This is a bias-variance tradeoff.

3.3.1 Regularized Discriminant Analysis

Regularized covariance matrix:

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}$$

where $\hat{\Sigma}$ is the pooled covariance matrix as used in LDA. α chosen in validation.

We can also shrink $\hat{\Sigma}$ toward the scalar matrix

$$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} + (1-\gamma)\hat{\sigma}^2 I$$

It leads to a more general family of covariance $\hat{\Sigma}(\alpha, \gamma)$.

3.3.2 Computations for LDA

Computation of LDA and QDA are simplified by diagonalizing $\hat{\Sigma}_k = U_k D_k U_k^T$. Then we can calculate LDA by $X* = D^{-\frac{1}{2}}U^T X$, which makes the covariance estimation identity, and classify to the closest class centroid in the transformed space, modulo the effect of the class prior probabilities π_k.

3.3.3 Reduced-Rank Linear Discriminant Analysis

The K centroids in p-dimensional space lie in an affine subspace of dimension $\leq K-1$. When p is much larger then K, there would be a large drop. Moreover, when locating the closest centroid, we may ignore directions orthogonal to the subspace since they contribute equally to the subspace. Thus we may as well project the X^* to the subspace-spanning subspace H_{K-1}, and make distance comparisons there. In doing so we would not have relinquished any of the information needed for LDA classification.

When $K > 3$, we might ask for a $L < K-1$-dimensional subspace H_L. Fisher defined optimal to mean that the projected centroids were spread out as much as possible in terms of variance, which means the PC subspaces of the centroids themselves. In summary, we can find the sequence of optimal subspaces for LDA by

- Compute $K \times p$ class centroids M and common covariance matrix W
- Compute $M^* = MW^{-\frac{1}{2}}$ using eigen-decomposition of W.
- Compute B^*, covariance matrix of M^* and its eigen-decomposition $V^* D_B V^{*T}$. Columns of V^* are the coordinates of the optimal subspace.

The lth discriminant variable is given by $Z_l = v_l^T X$ with $v_l = W^{-\frac{1}{2}} v_l^*$.

Fisher's approach: Find $Z = a^T X$ such that the between class variance is maximized relative to the within-class variance. The between class variance is the variance of the class means of Z, and the within class variance is the pooled variance about the means.

The between class variance is $a^T Ba$ and within class variance is $a^T Wa$. B is the covariance matrix of the class centroid matrix M. Total variance $T = B + W$. So Fisher's problem is actually maximizing Rayleigh quotient

$$\max_a \frac{a^T Ba}{a^T Wa}$$

We can easily get that a is given by the corresponding eigenvector of the largest eigenvalue of $W^{-1}B$. And $a_2, a_3 ... \ a_l$ are referred to as *discriminant coordinates/canonical variates.*

A Brief Summary

- Gaussian classification with common covariances leads to linear decision boundaries. Classification can be achieved by sphering the data with respect to W, and classifying to the closest centroid (modulo $\log \pi_k$) in the sphered space.
- Confine the data to the subspace spanned by the centroids in the sphered space.
- This subspace can be further decomposed into successively optimal subspaces in term of centroid separation. This decomposition is identical to the decomposition due to Fisher.

There is a close connection between Fisher's reduced rank discriminant analysis and regression of an indicator response matrix. A related fact is that if one transforms the original predictors X to $\hat{Y}$, then LDA using $\hat{Y}$ is identical to LDA in the original space followed by eigen-decomposition of $\hat{Y}^T Y$.

3.4 Logistic Regression

The model arises from the desire to model posterior probabilities of the K classes via linear functions and ensuring their sum is 1 and remain in $[0, 1]$.

$$\log \frac{Pr(G = i|X = x)}{Pr(G = K|X = x)} = \beta_{i0} + \beta_i^T x$$

It is specified in terms of $K - 1$ log-odds or logit transformations. Simple calculation shows that

$$\begin{aligned} \theta =& \{\beta_{10}, \beta_1^T, ..., \beta_{(K-1)0}, \beta_{K-1}^T\} \\ p_k(x, \theta) := Pr(G = k|X = x) =& \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)} \\ Pr(G = K|X = x) =& \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)} \end{aligned}$$

Logistic regression models are used mostly as a data analysis and inference tool, where the goal is to understand the role of the input variables in explaining the outcome. It is less sensitive to outliers compared with LDA.

3.4.1 Fitting Logistic Regression Models

Usually by maximum likelihood with conditional maximum likelihood of G given X, Since $Pr(G|X)$ specifies the conditional distribution, the multinormal distribution is appropriate. Log-likelihood for N observations is

$$l(\theta) = \sum_{i=1}^{N} log p_{g_i}(x_i; \theta), \quad p_k(x_i; \theta) = Pr(G = k|X = x_i; \theta)$$

For two-class case, let $y_i = g_i \in \{0, 1\}$, $p_1(x; \theta) = p(x; \theta)$, $\beta = \{\beta_{10}, \beta_1\}$,

$$\begin{aligned} l(\beta) =& \sum_{i=1}^{N} \{y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta))\} = \sum_{i=1}^{N} \{y_i \beta^T x_i - \log(1 + e^{\beta^T x_i})\} \\ \partial_\beta l(\beta) =& \sum_{i=1}^{N} x_i (y_i - p(x_i; \beta)) = 0 \end{aligned}$$

which are $p+1$ nonlinear equation of β. To solve this, we can use Newton-Raphson algorithm, where a single Newton update is

$$\begin{aligned}\beta^{\text{new}} =& \beta^{\text{old}} - \left(\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T}\right)^{-1}\partial_\beta l(\beta)\\ \frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T} =& -\sum_{i=1}^{N} x_i x_i^T p(x_i;\beta)(1-p(x_i;\beta))\end{aligned}$$

Matrix Form: Let p be the vector of $p(x_i;\beta^{\text{old}})$, $\mathbf{W}$ be a diagonal matrix with ith element $p(x_i;\beta^{old})(1-p(x_i;\beta^{old}))$

$$\begin{aligned}\frac{\partial \ell(\beta)}{\partial\beta} &= \mathbf{X}^T(\mathbf{y}-\mathbf{p})\\ \frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T} &= -\mathbf{X}^T\mathbf{W}\mathbf{X}\\ \beta^{\text{new}} &= \beta^{\text{old}} + \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{p})\\ &= \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}\left(\mathbf{X}\beta^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\right)\\ &= \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z}\\ \mathbf{z} &= \mathbf{X}\beta^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\end{aligned}$$

$\mathbf{z}$ is called *adjusted response*. Each iteration actually solves

$$\beta^{\text{new}} \leftarrow \arg\min_{\beta}(\mathbf{z}-\mathbf{X}\beta)^T\mathbf{W}(\mathbf{z}-\mathbf{X}\beta)$$

Log-likelihood is concave so it should converge, although overshooting can occur.

For $K \geq 3$, Newton algorithm can also be expressed as an iteratively reweighted least squares algorithm, but with a vector of $K-1$ responses and a nondiagonal weight matrix per observation.

3.4.2 Quadratic Approximations and Interface

- The weighted residual sum-of-squares is the Pearson chi-square statistic

$$\sum_{i=1}^{N}\frac{(y_i-\hat{p}_i)^2}{\hat{p}_i\,(1-\hat{p}_i)}$$

  a quadratic approximation to the deviance.

- If the model is correct, then maximum-likelihood parameter estimates $\hat{\beta}$ is consistent(converges to the true β).

- CLT shows that $\hat{\beta}$ converges to $N(\beta,(X^TWX)^{-1})$.

3.4.3 L_1 Regularized Logistic Regression

We would maximize

$$\max_{\beta_0,\beta}\left\{\sum_{i=1}^{N}\left[y_i\left(\beta_0+\beta^T x_i\right)-\log\left(1+e^{\beta_0+\beta^T x_i}\right)\right]-\lambda\sum_{j=1}^{p}\left|\beta_j\right|\right\}$$

We typically do not penalize the intercept term and standardize the predictors for the penalty to be meaningful. Lasso criterion is concave and can be solved by nonlinear programming methods. The score equations for non-zero coefficient variables are actually

$$\mathbf{x}_j^T(\mathbf{y}-\mathbf{p})=\lambda\cdot\operatorname{sign}\left(\beta_j\right)$$

3.4.4 Logistic Regression or LDA?

In LDA the linear log-posterior odds between class k and K is a consequence of the Gaussian assumption for densities and a common covariance matrix. The linear logistic by construction also has linear logits. Although they have the exact same form, the difference lies in the way the linear coefficients are estimated. The logistic regression model is more general, in that it makes less assumptions. The joint density of X and G is

$$\Pr(X,G=k)=\Pr(X)\Pr(G=k|X)$$

For both LDA and logistic, the second term is

$$\Pr(G=k|X=x)=\frac{e^{\beta_{k0}+\beta_k^T x}}{1+\sum_{\ell=1}^{K-1}e^{\beta_{\ell 0}+\beta_\ell^T x}}$$

Logistic model leaves $Pr(X)$ as an arbitrary density function. And fits parameters by maximizing conditional likelihood. Although $Pr(X)$ is ignored, we can think of this marginal density as being estimated in a fully nonparametric and unrestricted fashion, using the empirical distribution function which places mass $1/N$ at each observation.

With LDA we fit the parameters by maximizing full log-likelihood based on the joint density

$$\Pr(X,G=k)=\phi\left(X;\mu_k,\mathbf{\Sigma}\right)\pi_k$$

Here marginal density does play a role: it is the mixture density

$$\Pr(X)=\sum_{k=1}^{K}\pi_k\phi\left(X;\mu_k,\mathbf{\Sigma}\right)$$

which also involves parameters.

By relying on the additional model assumptions, we have more information about the parameters, and hence can estimate them more efficiently (lower variance).

However, observations far from the decision boundary (which are down-weighted by logistic regression) play a role in estimating the common covariance matrix, which means that LDA is not robust to gross outliers.

In practice these assumptions are never correct, and often some of the components of X are qualitative variables. It is generally felt that logistic regression is a safer, more robust bet than the LDA model, relying on fewer assumptions. It is our experience that the models give very similar results, even when LDA is used inappropriately, such as with qualitative predictors.

3.5 Separating Hyperplanes

Perceptron: Compute a linear combination of the input features and return the sign of $f(x) = \beta_0 + \beta^T x$. Separating hyperplane is $\beta_0 + \beta^T x = 0$.

For the line $\beta_0 + \beta^T x = 0$

- $\beta/||\beta||$ is the vector normal to the surface of the line.
- Signed distance of any point to the line is given by

$$beta^{*T}\left(x - x_0\right) = \frac{1}{\|\beta\|}\left(\beta^T x + \beta_0\right) = \frac{1}{\left\|f'(x)\right\|} f(x)$$

3.5.1 Rosenblatt's Perceptron Learning Algorithm

Target: minimizing the distance of misclassified points to the decision boundary, which is

$$\begin{aligned} D\left(\beta, \beta_0\right) &= -\sum_{i \in \mathcal{M}} y_i \left(x_i^T \beta + \beta_0\right) \\ \partial \frac{D\left(\beta, \beta_0\right)}{\partial \beta} &= -\sum_{i \in \mathcal{M}} y_i x_i \\ \partial \frac{D\left(\beta, \beta_0\right)}{\partial \beta_0} &= -\sum_{i \in \mathcal{M}} y_i \end{aligned}$$

Using SGD, a step is taken after each observation is visited,

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i x_i \\ y_i \end{pmatrix}$$

Here ρ is the learning rate. It will converge to a separating hyperplane in finite steps when classes are linear separable.

Problems:

- When the data are separable, there are many solutions, and which one is found depends on the starting values.
- The "finite" number of steps can be very large. The smaller the gap, the longer the time to find it.
- When the data are not separable, the algorithm will not converge, and cycles develop. The cycles can be long and therefore hard to detect.

3.5.2 Optimal Separating Hyperplanes

Target: Separates the two classes and maximizes the distance to the closest point from either class.

Optimization form:

$$\begin{gathered} \max_{\beta, \beta_0 = 1} M \\ \text{subject to } \frac{1}{\|\beta\|} y_i \left(x_i^T \beta + \beta_0\right) \geq M, i = 1, \ldots, N \end{gathered}$$

We can just set $\|\beta\| = 1/M$, then the problem becomes

$$\begin{gathered}\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 \\ \text{subject to } y_i\left(x_i^T\beta + \beta_0\right) \geq 1, i = 1, \ldots, N\end{gathered}$$

This is a convex problem(quadratic criterion with linear inequality constraints) with Lagrange function

$$L_P = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i\left(x_i^T\beta + \beta_0\right) - 1\right]$$

to be minimized with respect to β, β_0. Set derivatives to zero, we have

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i, \qquad 0 = \sum_{i=1}^{N} \alpha_i y_i$$

$$\text{(Wolfe Dual) } L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i\alpha_k y_i y_k x_i^T x_k \quad \text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

The solution is obtained by maximizing L_D in the positive orthant, a simpler convex optimization problem. In addition the solution must satisfy the Karush–Kuhn–Tucker conditions, which includes the above equations and

$$\alpha_i\left[y_i\left(x_i^T\beta + \beta_0\right) - 1\right] = 0 \quad \forall i$$

We can also see that β is defined as a linear combination of support points x_i-those points defined to be on the boundary of the slab via $\alpha_i > 0$.

The description of the solution in terms of support points seems to suggest that the optimal hyperplane focuses more on the points that count, and is more robust to model misspecification, while LDA depends on all of the data, even points far away from the decision boundary. However, the identification of these support points required the use of all the data. Of course, when the points are real Gaussian, LDA will be the optimal, and separating hyperplanes will pay a price for focusing on the (noisier) data at the boundaries of the classes.

When a separating hyperplane exists, logistic regression fit by maximum likelihood will always find it, since the log-likelihood can be driven to 0 in this case. Other common points include The coefficient vector is defined by a weighted least squares fit of a zero-mean linearized response on the input features, and the weights are larger for points near the decision boundary than for those further away.

Chapter 4

Basic Expansions And Regularization

4.1 Introduction

Core Idea: augment/replace the vector of inputs X with additional variables.

Let $h_m(X) : \mathbb{R}^p \to \mathbb{R}$ be the mth transformation,

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X)$$

A linear basis expansion in X. Once h_m determined, models are linear in these new variables. Examples include

- $h_m(X) = X_m$
- $H_m(X) = X_j^2$ or $H_m(X) = X_j X_k$. Enable us to augment inputs to achieve higher order Taylor expansions.
- $H_m(X) = \log(X_j)$, or other nonlinear transformations like $\|X\|$
- $H_m(X) = I(L_m \leq X_j \leq U_m$, indicator for a region

Let a dictionary $\mathcal{D}$ consists of all typical basis functions, we have **Three common approaches** to control model complexity

- Restriction methods, where we decide before-hand to limit the class of functions.
- Selection methods, which adaptively scan the dictionary and include only those basis functions h_m that contribute significantly to the fit of the model.
- Regularization methods where we use the entire dictionary but restrict the coefficients.

4.2 Piecewise Polynomials and Splines

Assume X is one-dimensional. A piecewise polynomial is representing f by different polynomials on different intervals. Let $h_m(X) = I(X \in [\xi_{m-1}, \xi_m])$, $f(X) = \sum_m \beta_m h_m(X)$. Least square estimate amounts to $\beta_m = \hat{Y}_m$, mean of Y on the mth region.

To make the function continuous, it requires $f(\xi_m^-) = f(\xi_m^+)$. A direct way is using basis $1, X, (X - \xi_m)_+$. For smoother function, we may increase the order of local polynomials.

Cubic Spline: Continuous and first/second derivatives continuous.

As for order-M spline with K knots ξ_j, it should have continuous derivatives up to order $M-2$. Its base is

$$h_j(X) = X^{j-1}, j = 1, \dots, M$$
$$h_{M+\ell}(X) = (X - \xi_\ell)_+^{M-1}, \ell = 1, \dots, K$$

Cubic spline: lowest-order spline without visible knot-discontinuity. **Seldom any good reason to go beyond that**.

4.2.1 Natural Cubic Splines

Behavior of polynomial fit to data may be erratic near boundaries, which can be dangerous and even more wild than global polynomials.

Natural Cubic Splines adds more constraints: function is linear beyond knots. A price in bias will be paid.

A natural cubic spline with K knots can be represented by K basis functions. One can start from a basis for cubic splines and reduce it by boundary constraints.

4.3 Filtering and Feature Extraction

Preprocess a high dimensional x to some new features by $x^* = g(x)$.

4.4 Smoothing Splines

Penalized residual sum of squares

$$\mathrm{RSS}(f, \lambda) = \sum_{i=1}^{N} \left\{ y_i - f(x_i) \right\}^2 + \lambda \int \left\{ f''(t) \right\}^2 dt \tag{4.1}$$

λ: fixed smoothing parameter. Here the first term measures closeness of the data and the second penalizes curvature.

Criterion (4.1) is defined on a Sobolev space, which is a space of functions for which the second term is defined. It can be shown that it has an explicit unique finite-dimensional minimizer with a natural spline with N knots. Since the solution is a natural spline, we can write it as

$$f(x) = \sum_{j=1}^{N} N_j(x)\theta_j$$

Here $N_j(x)$ is an N-dimensional set of basis for representing this family of natural splines. The criterion then reduces to

$$\mathrm{RSS}(\theta, \lambda) = (\mathbf{y} - \mathbf{N}\theta)^T(\mathbf{y} - \mathbf{N}\theta) + \lambda\theta^T\mathbf{\Omega}_N\theta$$

where $\{\mathbf{N}_{ij} = N_j(x_i)\}$ and $\{\Omega_N\}_{jk} = \int N_j''(t)N_k''(t)dt$. The solution seems to be

$$\hat{\theta} = \left(\mathbf{N}^T\mathbf{N} + \lambda\mathbf{\Omega}_N\right)^{-1}\mathbf{N}^T\mathbf{y}$$

a generalized ridge regression, and fitted spline is given by

$$\hat{f}(x) = \sum_{j=1}^{N} N_j(x)\hat{\theta}_j$$

4.4.1 Degrees of Freedom and Smoother Matrices

Here we discuss intuitive ways of prespecifying λ.

A smooth spline with prechosen λ is a linear smoother since $\hat{\theta}$ is a linear combination of y. Let $\hat{\mathbf{f}}$ be the estimated values at x_i,

$$\hat{\mathbf{f}} = \mathbf{N}\left(\mathbf{N}^T\mathbf{N} + \lambda\mathbf{\Omega}_N\right)^{-1}\mathbf{N}^T\mathbf{y} = \mathbf{S}_\lambda\mathbf{y}$$

$\mathbf{S}_\lambda$ known as smoother matrix.

Let $\mathbf{B}_\xi$ be $N \times M$ matrix of M spline basis functions at x_i. $M \ll N$. Then least squares fitted spline value is defined by

$$\hat{\mathbf{f}} = \mathbf{B}_\xi\left(\mathbf{B}_\xi^T\mathbf{B}_\xi\right)^{-1}\mathbf{B}_\xi^T\mathbf{y} = \mathbf{H}_\xi\mathbf{y}$$

$\mathbf{H}_\xi$ is a project operator(hat matrix). $M = tr(\mathbf{H}_\xi)$ gives the dimension of the projection space, which is also the number of basis functions/parameters involved in the fit. By analogy we define *effective degrees of freedom of a smoothing spline* is

$$\mathrm{df}_\lambda = \mathrm{trace}\left(\mathbf{S}_\lambda\right)$$

We can then find λ by specifying df_λ.

Since $\mathbf{S}_\lambda$ is symmetric, it has a real decomposition, and we can rewrite it in Reinsch form

$$\mathbf{S}_\lambda = (\mathbf{I} + \lambda\mathbf{K})^{-1}$$

where $\mathbf{K}$ does not depend on λ. $\hat{\mathbf{f}} = \mathbf{S}_\lambda\mathbf{y}$ actually solves

$$\min_{\mathbf{f}}(\mathbf{y}-\mathbf{f})^T(\mathbf{y}-\mathbf{f}) + \lambda\mathbf{f}^T\mathbf{K}\mathbf{f}$$

$\mathbf{K}$ known as a penalty matrix. Let d_k be the kth eigenvalue of $\mathbf{K}$, the eigen-decomposition of $\mathbf{S}_\lambda$ is

$$\mathbf{S}_\lambda = \sum_{k=1}^N \rho_k(\lambda)\mathbf{u}_k\mathbf{u}_k^T = \sum_{k=1}^N \frac{1}{1+\lambda d_k}\mathbf{u}_k\mathbf{u}_k^T$$

By eigen-representation, we can also find that

- Eigenvectors not affected by changes in λ.
- $\mathbf{S}_{\lambda\mathbf{Y}} = \sum_{k=1}^N \mathbf{u}_k\rho_k(\lambda)\langle\mathbf{u}_k, \mathbf{y}\rangle$, hence the smoothing spline operates by decomposing $\mathbf{y}$ w.r.t $\{\mathbf{u}_k\}$ and differentially shrinking the contributions using $\rho_k(\lambda)$, while $\mathbf{H}_\xi$ has M eigenvalues equal to 1 and the rest are 0. So smoothing splines are shrinking smoothers, regression splines are projection smoothers.
- $\mathbf{u}_k$ ordered by decreasing $\rho_k(\lambda)$ increase in complexity. Higher-complexity $\mathbf{u}_k$ are shrunk more. If domain of X are periodic, $\mathbf{u}_k$ will be sin/cos.
- First two eigenvalues always 1(two-dimensional eigenspace of functions linear in x).
- $d_1 = d_2 = 0$ and linear functions are not penalized.
- Reparametrizing using Demmler-Reinsch basis $\mathbf{u}_k$ solves

$$\min_{\boldsymbol{\theta}}\|\mathbf{y} - \mathbf{U}\boldsymbol{\theta}\|^2 + \lambda\boldsymbol{\theta}^T\mathbf{D}\boldsymbol{\theta}, \quad \mathbf{D} = diag\{d_k\}$$

A smoothing spline is a local fitting method, much like the locally weighted regression. As $\lambda \to 0$, $\mathrm{df}_\lambda \to N$, $\mathbf{S}_\lambda \to \mathbf{I}$, As $\lambda \to \infty$, $\mathrm{df}_\lambda \to 2$, $\mathbf{S}_\lambda \to \mathbf{H}$, the hat matrix for linear regression.

4.5 Automatic Selection of the Smoothing Parameters

4.5.1 Fixing the Degree of Freedom

$\mathrm{df}_\lambda = \mathrm{trace}\left(\mathbf{S}_\lambda\right)$ is monotone in λ. We can set df_λ to fix λ. We may try some different df and use some traditional criteria like F-tests, residual plots or others.

4.5.2 The Bias-Variance Tradeoff

Since $\hat{\mathbf{f}} = \mathbf{S}_\lambda \mathbf{y}$, let $\mathrm{Cov}(\mathbf{y}) = I$

$$\begin{aligned}\mathrm{Cov}(\hat{\mathbf{f}}) =& \mathbf{S}_\lambda \mathrm{Cov}(\mathbf{y}) \mathbf{S}_\lambda^T = \mathbf{S}_\lambda \mathbf{S}_\lambda^T \\ \mathrm{Bias}(\hat{\mathbf{f}}) =& \mathbf{f} - \mathrm{E}(\hat{\mathbf{f}}) = \mathbf{f} - \mathbf{S}_\lambda \mathbf{f}\end{aligned}$$

where $\mathbf{f}$ is the (unknown) vector of evaluations of the true f.

The integrated squared prediction error (EPE) combines both bias and variance in a single summary:

$$\begin{aligned}\mathrm{EPE}\left(\hat{f}_\lambda\right) &= \mathrm{E}\left(Y - \hat{f}_\lambda(X)\right)^2 = \mathrm{Var}(Y) + \mathrm{E}\left[\mathrm{Bias}^2\left(\hat{f}_\lambda(X)\right) + \mathrm{Var}\left(\hat{f}_\lambda(X)\right)\right] \\ &= \sigma^2 + \mathrm{MSE}\left(\hat{f}_\lambda\right)\end{aligned}$$

EPE is a natural quantity of interest, and does create a tradeoff between bias and variance. When we don't know the true function, we do not have access to EPE, and need an estimate. Overall $N - fold$ cross-validation curve could be a good estimation of EPE.

4.6 Nonparametric Logistic Regression

Consider a general logistic model

$$\begin{aligned}\log \frac{\Pr(Y=1|X=x)}{\Pr(Y=0|X=x)} &= f(x) \\ \Pr(Y=1|X=x) &= \frac{e^{f(x)}}{1+e^{f(x)}}\end{aligned}$$

To fit $f(x)$ in a smooth fashion, consider the penalized log-likelihood criterion

$$\begin{aligned}\ell(f;\lambda) &= \sum_{i=1}^N \left[y_i \log p\left(x_i\right) + \left(1-y_i\right) \log\left(1 - p\left(x_i\right)\right)\right] - \frac{1}{2}\lambda \int \left\{f''(t)\right\}^2 dt \\ &= \sum_{i=1}^N \left[y_i f\left(x_i\right) - \log\left(1 + e^{f(x_i)}\right)\right] - \frac{1}{2}\lambda \int \left\{f''(t)\right\}^2 dt\end{aligned}$$

We can also represent $f(x) = \sum_{j=1}^N N_j(x)\theta_j$, and

$$\begin{aligned}\frac{\partial \ell(\theta)}{\partial \theta} &= \mathrm{N}^T(\mathrm{y} - \mathrm{p}) - \lambda \Omega \theta \\ \frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta^T} &= -\mathrm{N}^T \mathrm{W} \mathrm{N} - \lambda \Omega\end{aligned}$$

where W is a diagonal matrix of $p(x_i)(1-p(x_i))$. The first derivative is nonlinear in θ, using Newton-Rapthson in method, we can get update equation

$$\begin{aligned}\theta^{\text{new}} &= \left(\mathbf{N}^T\mathbf{W}\mathbf{N}+\lambda\mathbf{\Omega}\right)^{-1}\mathbf{N}^T\mathbf{W}\left(\mathbf{N}\theta^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\right) = \left(\mathbf{N}^T\mathbf{W}\mathbf{N}+\lambda\mathbf{\Omega}\right)^{-1}\mathbf{N}^T\mathbf{W}\mathbf{z}\\ \mathbf{f}^{\text{new}} &= \mathbf{N}\left(\mathbf{N}^T\mathbf{W}\mathbf{N}+\lambda\mathbf{\Omega}\right)^{-1}\mathbf{N}^T\mathbf{W}\left(\mathbf{f}^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\right) = \mathbf{S}_{\lambda,w}\mathbf{z}\end{aligned}$$

see that the update fits a weighted smoothing spline to the working response $\mathbf{z}$.

4.7 Multidimensional Splines

Now we consider multi-dimensional X. Let $X \in \mathbb{R}^2$, then the $M_1 \times M_2$ dimensional *tensor product basis* is defined by

$$\begin{aligned}g_{jk}(X) =& h_{1j}(X_1)\, h_{2k}(X_2), j = 1,\ldots,M_1, k = 1,\ldots,M_2\\ g(X) =& \sum_{j=1}^{M_1}\sum_{k=1}^{M_2}\theta_{jk}g_{jk}(X)\end{aligned}$$

Smoothing spline via regularization can also be constructed by

$$\min_f \sum_{i=1}^{N}\left\{y_i - f(x_i)\right\}^2 + \lambda J[f]$$

whose solution has the form

$$f(x) = \beta_0 + \beta^T x + \sum_{j=1}^{N}\alpha_j h_j(x)$$

where $h_j(x) = \|x-x_i\|^2 \log\|x-x_i\|$ are radial basis functions.

Complexity would be $O(N^3)$. In practice, it is usually sufficient to work with K knots covering the domain. , which reduces the computations to $O(NK^2+K^3)$.

Additive spline models are a restricted class of splines where $f(X) = \alpha + \sum f_i(X_i)$, which might also be more natural, and can be extended to ANOVA spline decompositions.

$$f(X) = \alpha + \sum_j f_j(X_j) + \sum_{j<k} f_{jk}(X_j, X_k) + \cdots$$

Things to consider:

- Maximum order of iteraction
- Terms to include
- Representation: Groups of small basis versus a large complete basis

4.8 Regularization and Reproducing Kernel Hilbert Spaces

Cast splines into the larger context of regularization methods and reproducing kernel Hilbert spaces.

A general class has the form

$$\min_{f\in\mathcal{H}}\left[\sum_{i=1}^{N}L\left(y_i,f\left(x_i\right)\right)+\lambda J(f)\right]$$

where $\mathcal{H}$ is a space where $J(f)$ is defined. A general penalty functional has the form

$$J(f)=\int_{\mathbb{R}^d}\frac{|\tilde{f}(s)|^2}{\tilde{G}(s)}ds$$

where $\tilde{f}$ is a Fourier transform of f, and $\tilde{G}$ is some positive function which falls to zero as $\|s\|\to\infty$. Under some additional assumptions, the solutions have the form

$$f(X)=\sum_{k=1}^{K}\alpha_k\phi_k(X)+\sum_{i=1}^{N}\theta_i G\left(X-x_i\right)$$

where ϕ_k span the null space of the penalty function J, and G is the inverse Fourier transform of $\tilde{G}$.

4.8.1 Spaces of Functions Generated by Kernels

An important subclass of the general regularization problems are generated by positive definite kernel $K(x,y)$, and the corresponding Hilbert space $\mathcal{H}_K$ is called a *reproducing kernel Hilbert space*(RKHS). J is defined in terms of kernels as well.

Let $x,y\in\mathbb{R}^p$, we consider the space of functions generated by the span of $\{K(\cdot,y),y\in\mathbb{R}^p\}$. Suppose K has an eigen-expansion

$$K(x,y)=\sum_{i=1}^{\infty}\gamma_i\phi_i(x)\phi_i(y),\quad \gamma_i\geq 0,\sum_{i=1}^{\infty}\gamma_i^2<\infty$$

Elements of $\mathcal{H}_K$ have an expansion in terms pf eigen functions

$$f(x)=\sum_{i=1}^{\infty}c_i\phi_i(x),\quad J(f):=\|f\|_{\mathcal{H}_K}^2\overset{\text{def}}{=}\sum_{i=1}^{\infty}c_i^2/\gamma_i<\infty$$

It is a generalized ridge penalty, and functions with larger eigenvalues get penalized less.

Rewriting the general form we have

$$\min_{f\in\mathcal{H}_K}\left[\sum_{i=1}^{N}L\left(y_i,f\left(x_i\right)\right)+\lambda\|f\|_{\mathcal{H}_K}^2\right]=\min_{\{c_j\}_1^\infty}\left[\sum_{i=1}^{N}L\left(y_i,\sum_{j=1}^{\infty}c_j\phi_j\left(x_i\right)\right)+\lambda\sum_{j=1}^{\infty}c_j^2/\gamma_j\right]$$

It can be shown that the solution is finite-dimensional and has the form

$$f(x)=\sum_{i=1}^{N}\alpha_i K\left(x,x_i\right)$$

The basis function $h_i(x)=K(x,x_i)$ is the *representer of evaluation* at x_i in $\mathcal{H}_K$, for $f\in\mathcal{H}_K$, $\left\langle K\left(\cdot,x_i\right),f\right\rangle_{\mathcal{H}_K}=f\left(x_i\right)$, $\left\langle K\left(\cdot,x_i\right),K\left(\cdot,x_j\right)\right\rangle_{\mathcal{H}_K}=K\left(x_i,x_j\right)$, hence

$$J(f)=\sum_{i=1}^{N}\sum_{j=1}^{N}K\left(x_i,x_j\right)\alpha_i\alpha_j\text{ for }f(x)=\sum_{i=1}^{N}\alpha_i K\left(x,x_i\right)$$

The finite-dimensional criterion of the general form is then

$$\min_{\alpha} L(\mathbf{y}, \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$$

Bayesian interpretation: f is a zero-mean stationary Gaussian with prior covariance K.

A general case is $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$ where null space $\mathcal{H}_0$ consisting of functions that do not get penalized. The penalty becomes $J(f) = \|P_1 f\|$ where P_1 is the orthogonal projection onto $\mathcal{H}_1$. The solution then has the form $f(x) = \sum_{j=1}^{M} \beta_j h_j(x) + \sum_{i=1}^{N} \alpha_i K(x, x_i)$. From a Bayesian perspective, the components in $\mathcal{H}_0$ have improper priors(with variance).

4.8.2 Examples of RKHS

$$\min_{\alpha} (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha})^T (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$$

Penalized Polynomial Regression

The kernel $K(x, y) = (\langle x, y \rangle + 1)^d$, for $x, y \in \mathbb{R}^p$, has C_{p+d}^d eigen-functions that span the space of d-dim polynomials in $\mathbb{R}^p$.

Gaussian Radial Basis Functions

Gaussian kernel $K(x, y) = e^{-\nu \|x-y\|^2}$ is chosen because of its functional form in the representation $f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i)$. Gaussian radial basis functions are

$$k_m(x) = e^{-\nu \|x - x_m\|^2}, m = 1, \ldots, N$$

The coefficients are estimated with $\min_{\alpha} (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha})^T (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$.

Support Vector Classifiers

Supporting vector machines $f(x) = \alpha_0 + \sum_{i=1}^{N} \alpha_i K(x, x_i)$ where the parameters are chosen to minimize

$$\min_{\alpha_0, \alpha} \left\{ \sum_{i=1}^{N} \left[1 - y_i f(x_i) \right]_+ + \frac{\lambda}{2} \alpha^T \mathbf{K} \alpha \right\}$$

where $y_i \in \{0, 1\}$. It is a quadratic optimization problem with linear constraints.

4.9 Wavelet Smoothing

Wavelets typically use a complete orthonormal basis to represent functions, but then shrink and select the coefficients toward a sparse representation. They are able to represent both smooth and/or locally bumpy functions in an efficient way—a phenomenon dubbed time and frequency localization. In contrast, the traditional Fourier basis allows only frequency localization.

4.9.1 Wavelet Bases and the Wavelet Transform

Bases are generate by translations and dilations of a single scaling function $\phi(x)$(*father*). For example, consider Haar basis, space V_j spanned by $\phi_{j.k} = 2^{j/2}\phi(2^j x - k)$, $\phi(x) = I(x \in [0,1])$. $\cdots \supset V_1 \supset V_0 \supset V_{-1} \supset \cdots$ (V_0 the reference space V_0).

As for the definition of wavelets, we want to represent a function in V_{j+1} by a component in V_j plus in the orthogonal complement W_j of V_j to V_{j+1}, written as $V_{j+1} = V_j \oplus W_j$. W_j represents detail, and we might wish to set some elements of it to zero. Functions $\psi(x-k)$ generated by the mother wavelet $\psi(x) = \phi(2x) - \phi(2x-1)$ form an orthonormal basis for W_0 for the Haar family, and $\psi_{j,k} = 2^{j/2}\psi\left(2^j x - k\right)$ form a basis for W_j. We also know $V_J = V_0 \oplus W_0 \oplus W_1 \cdots \oplus W_{J-1}$.

Since the spaces are orthogonal, all basis functions should be orthonormal.

Tradeoff between Wavelet Basis

- The wider the support, the more time the wavelet has to die to zero, so it can achieve this more smoothly. Note that the effective support seems to be much narrower.
- V_0 is equivalent to the null space of the smoothing-spline penalty.

4.9.2 Adaptive Wavelet Filtering

Wavelets are very useful when the data are measured on a uniform lattice(discretized signal, image, or time series). Here we consider one-dimensional case with $N = 2^J$ lattice-points. Let $\mathbf{W}$ be the $N \times N$ orthonormal wavelet basis matrix evaluated at the N uniformly spaced observations, $\mathbf{y}$ be the response vector, Then $\mathbf{y}^* = \mathbf{W}\mathbf{y}$ is the *wavelet transform* of $\mathbf{y}$. A popular method for adaptive wavelet fitting is SURE shrinkage. We start with

$$\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{W}\boldsymbol{\theta}\|_2^2 + 2\lambda\|\boldsymbol{\theta}\|_1$$

which is same as the lasso criterion. Since $\mathbf{W}$ is orthonormal,

$$\hat{\theta}_j = \text{sign}\left(y_j^*\right)\left(\left|y_j^*\right| - \lambda\right)_+$$

The fitted function (vector) is then given by the inverse wavelet transform $\hat{\mathbf{f}}^* = \mathbf{W}\hat{\mathbf{y}}$.

A simple choice for λ is $\lambda = \sigma\sqrt{2\log N}$($\sigma$ the estimate of the standard deviation). The reason is when $\mathbf{y}$ are white noise, so are $\mathbf{y}^*$. This λ is the expected maximum of N white noise with variance σ^2.

Chapter 5

Model Assessment and Selection

Key methods for performance assessment used to select models.

5.1 Bias, Variance and Model Complexity

Test error/generalization error given training set $\mathcal{T}$ is defined as

$$\mathrm{Err}_{\mathcal{T}} = \mathrm{E}[L(Y, \hat{f}(X))|\mathcal{T}]$$

And expected prediction error is $\mathrm{Err} = \mathrm{E}[L(Y, \hat{f}(X))] = \mathrm{E}\left[\mathrm{Err}_{\mathcal{T}}\right]$.

Estimation of $\mathrm{Err}_{\mathcal{T}}$ is our goal, but Err is more amenable to statistical analysis. It does not seem possible to efficiently estimate conditional error only given the information in the same training set.

Training error is defined as

$$\overline{\mathrm{err}} = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \hat{f}\left(x_i\right)\right)$$

When model becomes more complex, it uses training data more and is able to adapt more complicated structures, with a decrease in bias and increase in variance. Unfortunately, training error could not be a good estimation of test error.

Typical loss function for classification functions includes

$$L(G, \hat{G}(X)) = I(G \neq \hat{G}(X)) \quad (0-1 \text{ loss })$$

$$L(G, \hat{p}(X)) = -2\sum_{k=1}^{K} I(G=k)\log \hat{p}_k(X) = -2\log \hat{p}_G(X) \quad (-2\times \text{ log-likelihood })$$

G the categorical response, $p_k(X) = Pr(G=k|X)$, $\hat{G}(X) = \arg\max_k \hat{p}_k(X)$. $-2\times$ log-likelihood is sometimes referred to as the deviance. Log-likelihood can be used as a loss function for general response densities such as the Poisson, gamma, exponential, log-normal and others. "-2" makes the log-likelihood loss for the Gaussian distribution match squared-error loss.

Two steps

- **Model Selection**: estimating performance of different models to find the best.
- **Model assessment**: having chosen a final model, estimating its generalization error on new data.

In data rich situation: randomly divide training/validation/test set. A typical split could be 50/25/25.

5.2 The Bias-Variance Decomposition

If we assume $Y = f(X) + \epsilon$, $E(\epsilon) = 0$, $Var(\epsilon) = \sigma_\epsilon^2$, we can derive

$$\begin{aligned}
\operatorname{Err}(x_0) &= E\left[\left(Y - \hat{f}(x_0)\right)^2 | X = x_0\right] \\
&= \sigma_\varepsilon^2 + \left[\mathrm{E}\hat{f}(x_0) - f(x_0)\right]^2 + E\left[\hat{f}(x_0) - \mathrm{E}\hat{f}(x_0)\right]^2 \\
&= \sigma_\varepsilon^2 + \operatorname{Bias}^2\left(\hat{f}(x_0)\right) + \operatorname{Var}\left(\hat{f}(x_0)\right) \\
&= \text{ Irreducible Error } + \operatorname{Bias}^2 + \text{ Variance.}
\end{aligned}$$

For kNN fit, it has the simple form

$$\operatorname{Err}(x_0) = E\left[\left(Y - \hat{f}_k(x_0)\right)^2 | X = x_0\right] = \sigma_\varepsilon^2 + \left[f(x_0) - \frac{1}{k}\sum_{\ell=1}^{k} f\left(x_{(\ell)}\right)\right]^2 + \frac{\sigma_\varepsilon^2}{k}$$

For linear model $\hat{f}_p(x) = x^T\hat{\beta}$, we have

$$\operatorname{Err}(x_0) = E\left[\left(Y - \hat{f}_p(x_0)\right)^2 | X = x_0\right] = \sigma_\varepsilon^2 + \left[f(x_0) - \mathrm{E}\hat{f}_p(x_0)\right]^2 + \left\|\mathbf{h}(x_0)\right\|^2 \sigma_\varepsilon^2$$

where $\mathbf{h}(x_0) = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1} x_0$, the N vector of linear weight that produce the fit $\hat{f}_p(x_0) = x_0^T\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$, and hence $\operatorname{Var}\left[\hat{f}_p(x_0)\right] = \left\|\mathbf{h}(x_0)\right\|^2\sigma_\varepsilon^2$. While the variance change with x_0, its average is $(p/N)\sigma_\epsilon^2$, and hence

$$\frac{1}{N}\sum_{i=1}^{N}\operatorname{Err}(x_i) = \sigma_\varepsilon^2 + \frac{1}{N}\sum_{i=1}^{N}\left[f(x_i) - \mathrm{E}\hat{f}(x_i)\right]^2 + \frac{p}{N}\sigma_\varepsilon^2$$

The in-sample error. Effected by number of parameters p.

Ridge's Err is overall similar with $\mathbf{h}(x_0) = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \alpha\mathbf{I}\right)^{-1} x_0$. The bias will also be different. For a linear model, we can break down the bias term finely. Let

$$\beta_* = \arg\min_\beta \mathrm{E}\left(f(X) - X^T\beta\right)^2$$

Expectation taken with respect to X distribution, and

$$\begin{aligned}
\mathrm{E}_{x_0}\left[f(x_0) - \mathrm{E}\hat{f}_\alpha(x_0)\right]^2 &= \mathrm{E}_{x_0}\left[f(x_0) - x_0^T\beta_*\right]^2 + \mathrm{E}_{x_0}\left[x_0^T\beta_* - \mathrm{E}x_0^T\hat{\beta}_\alpha\right]^2 \\
&= \text{ Ave[Model Bias] }^2 + \text{ Ave [Estimation Bias] }^2
\end{aligned}$$

For linear models fit by OLS, the estimation bias is zero. For restricted fits, such as ridge regression, it is positive, and we trade it off with the benefits of a reduced variance. The model bias can only be reduced by enlarging the class of linear models to a richer collection of models, by including interactions and transformations of the variables in the model.

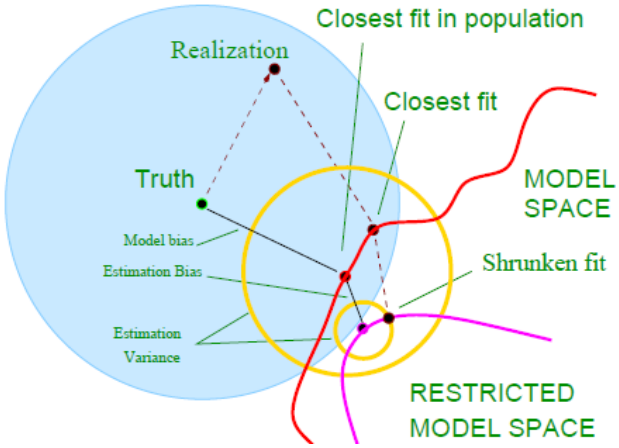

Figure 5.1: Bias-Variance Relationship

5.2.1 Optimism of the Training Error Rate

Chapter 6

Unsupervised Learning

6.1 Principle Components, Curves and Surfaces

6.1.1 Principle Components

Principle Components: A sequence of projections of the data, mutually uncorrelated and ordered in variance. Presented as linear manifolds.

Given observations by $x_1, x_2, ..., x_N$, consider the rank-q linear model:

$$f(\lambda) = \mu + V_q\lambda \tag{6.1}$$

$\mu \in \mathbb{R}^p, V^q \in \mathbb{R}^{p\times q}, \lambda \in \mathbb{R}^q$. To fit this model, we need to optimize the reconstruction error:

$$\min_{\mu,\lambda_i,V_q} \sum_N ||x_i - \mu - V_q\lambda_i||^2 \tag{6.2}$$

while it can be calculated by partially optimization that $\hat{\mu} = \bar{x}, \hat{\lambda_i} = V_q^T(x_i - \bar{x})$, then we can convert (6.2) to $\min_{V_q} \sum_N ||(x_i - \bar{x}) - V_qV_q^T(x_i - \bar{x})||^2$. W.L.O.G, we can assume $\bar{x} = 0$, and $H_q = V_qV_q^T$ is actually a projection matrix.

Now consider the singular value decomposition $X = UDV^T$, where $U \in \mathbb{R}^{N\times p}, I \in \mathbb{R}^{p\times p}$ orthogonal, D diagonal. $\forall q$, V_q consists of the first q columns of V, and the columns of UD are called the principle components. The N optimal $\hat{\lambda_i}$ is actually the first q principle components.

Besides, Xv_i has the highest variance among all linear combinations of the features orthogonal to $v_1, v_2, ..., v_{i-1}$

Ex 6.1.1. Handwritten Digits Recognition (P536)

Ex 6.1.2. Procrustes Transformation and Shape Averaging(P539): in this problem we use Frobenius Norm $||X||_F^2 = trace(X^TX)$

Principle Curves and Surfaces

Orientation: Generalize principle component to curved manifold approximation.