STAT 444 Statistical Learning

Austin Xia

July 26, 2020

Contents

List of Figures

List of Tables

1 Course Information

1.1 Contact

Instructor: Reza Ramezan
Email: rramezan@uwaterloo.ca

1.2 Grade

Assignments 70%
Quizzes 30%

2 Global modelling methods

2.1 Quick review of linear regression

2.1.1 Simple Linear Regression

$$\mu(x_i) = \mu_{0i} + \mu_{1i}(x_i)$$

regression line is

$$\hat{\mu}(x_i) = \hat{\mu}_{0i} + \hat{\mu}_{1i}(x_i)$$

$y - \mu(x_i)$ and $y - \hat{\mu}(x_i)$ are not the same thing

2.1.2 Multiple Linear Regression

$y = \mu(x) + r$ where $\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ here comes p value and t score

$$Y|X = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$$

Y is response vector
β is parameter
ϵ is error terms
X is design matrix

2.1.3 Least squares estimation

Definition 2.1.1 (RSS)

$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 = (y - X\beta)^T (y - X\beta)$$

SSE and RSS are the same thing, both referring to the Residuals sum of squares

Definition 2.1.2 (Ordinary least square estimate)

$$\hat{\beta} = argmin_{\beta}(RSS(\beta)) = argmin\left((y - X\beta)^T(y - X\beta)\right)$$

$$\frac{d}{d\beta}RSS(\beta) = -2X^T(y - X\beta)$$

$$X^T(y - X\beta) = 0$$

$$X^T y = X^T X\beta$$

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

It's normally asuumed that:

- The mean of Y is linear function of X
- Error terms have constant mean 0
- Error terms have constant variance σ^2
- Error terms follow a normal distribution
- Error terms are independent

Need these assumptions to do prediction

Theorem 2.1.1
Assume we can write y in terms of its mean and an error term

$$Y = E(Y|X) + \epsilon = \beta_0 + \sum \beta_i X_i + \epsilon$$

where $\epsilon \sim MVN(\beta, \sigma^2 I_{n*n})$ then

$$\tilde{\beta}_{OLS} \sim MVN(\beta, \sigma^2(X^T X)^{-1})$$

$$(n - p - 1)\frac{\tilde{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1}$$

Definition 2.1.3 (The Hat Matrix)

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

We have:

- $H^2 = H$
- H(1-H)=1
- tr(H)=p+1

$\hat{r}$ verses ϵ

- $\epsilon = y - X\beta$
- $r = y - \hat{y} = y - X\hat{\beta}$
- r is observed residual, ϵ is model error terms
- We will use $\hat{r}$ as approximate values for ϵ to check if
  - $E(\epsilon_i) = 0$
  - $Var(\epsilon_i) = \sigma^2$
  - $\epsilon_1, ..., \epsilon_n$ are Normally distrubuted
  - $\epsilon_1, ..., \epsilon_n$ are independent

2.1.4 Alternative methods to ordinary least squares regression

If assumptions on error term don't hold true:

- Use transformation to project onto a space where assumption hold true
- Use weighted least squares regression
- Use non-parametric models. e.g. kernel regression, splines, etc.

2.1.5 Influential Points

Based on hat matrix H $\hat{y} = Hy$, we measure influnce by:

$$\frac{d\hat{y}_i}{dy_i} = H_{ii}$$

which is influnence of y_i on the data.
If high leverage points exist in data, one can use robust regression model
Multicolinearity, Curse of Dimensionality $(X^TX)^{-1}$ must exsit, so under multicollinearity or $p \geq n$, LS estimation breaks down.
or if rows are closed to linearly dependent, trace of Hat Matrix get to large, causing prediction having variance too large

2.1.6 Reducible vs. irreducible errors

- prediction is for unobserved data, it needs test set
- inference. We would like to answer these questions in light of sampling variability.

$$MSE = E\left([Y - \hat{Y}]^2|X\right) = \left(f(x) - \hat{f}(x)\right)^2 + Var(\epsilon)$$

the first term is Reducible error, could be 0 if $\hat{f} == f$
the second term is irreducible error. caused by projecting Y into space of X

2.1.7 Bias-Variance tradeoff

$$E\left([Y_0 - \hat{f}(x_0)]^2\right) = E[(f(x_0) - E(\hat{f}(x_0))]^2 + E[(\hat{f}(x_0) - E(\hat{f}(x_0))]^2 + Var(\epsilon)$$

which is

$$Bias(\hat{f}(x_0))^2 + Var(\hat{f}(x_0))^2 + Var(\epsilon)$$

Bias decrease, flexibility increase as complexity increase
Variance increase, stability decrease as complexity increase

2.1.8 Cross-Validation: Basic idea

Definition 2.1.4 (Expected error)
$E(L(Y, \hat{f}(X))$ where $\hat{f}$ is estimate of f, L (loss function) measures distance between Y and $\hat{f}(X)$
eg squared error loss function is defined as

$$L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2$$

Cross Validation forms many test/training sets and measure the prediction error on each test set. The average of these error setimates the expected prediction error

Definition 2.1.5

- Partition data ramdomly to k disjoint and equal-size sets $T_1, ..., T_n$ each of size $n_k = n/k$
- For $i = 1, ..., k$ construct training sets $T_{-i} = T_1 ... \cup T_{i-1} \cup T_{i+1} \cup ...T_n$
  calculate
  $$sMSE_i = \frac{1}{n_k} \sum_{j \in T_i} \left(y_j - \hat{f}^{-i}(x_j) \right)^2$$
  where $f^{-i}(x_j)$ is estimate/fitted value of y_i based on a model fitted to T_{-i}
- the overall k-fild cross-validation error is
  $$CV_k = \frac{1}{k} \sum_{i=1}^{k} sMSE_i$$

Note: loss function can be replaced
it can be shown that

$$CV_k = \frac{1}{n} \sum_{j=1}^{n} \left(y_j - \hat{f}^{-\omega(j)}(x_j) \right)^2$$

this formula is understood base on test set
$\omega : 1...n \to 1...k$ is an indexing function that inducates the partition to which observation j is allocated by the ramdomization
choice of k is bias-variance trade-off. Large k results in similar training sets, high variance, smaller bias, more flexibility
in practice, k is set to 5/10

2.1.9 Leave-One-Out CV: LOOCV

- k=n in k-fold cross validation. i.e. training sets of size n-1 and test sets of size 1
- it has largest k, so very little bias and large variance low prediction power.
- it's computationally efficient. For least squres regression, it can be shown that
  $$CV_n = \frac{1}{n} \sum_{i=1}^{n} sMSE_i = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y_i}}{1 - H_{ii}} \right)^2$$
  The result above is true for most linear smoothers $\hat{Y} = SY$ under squared error loss, in which case H_{ii} will be repalced by S_{ii}

2.1.10 Generalized Cross-Validation:GCV

A convenient approcimation to LOOCV for linear fitting under squared-error loss. The GCV approximation of the prediction error is

$$GCV_n = \frac{1}{n} \sum \left(\frac{y_i - \hat{f}(x_i)}{1 - trace(S)/n} \right)$$

trace(S) is effective number of parameters in the model (degree of freedom). so number of parameter increase then GCV decrease prediction power

2.1.11 Best-subset selection

The subset selection is the process of selecting q ¡ p explanatory variates in modelling the function f
Problem with LS estimate $X\hat{\beta} = HY$

- prediction accurary, LS have low bias, but if not $n \gg n$ they have larger variance.
- multicollinearity
- if $p > n$ then LS estimate are not unique
- Model interpretability: including irrelevant variables leads to unneccessary complexity

Best-subset Selection:

- a total of $2^n = \binom{p}{0} + ... + \binom{p}{p}$ models are to be fitted
- An efficient algorithm makes this feasible for p as large as 30-40

Steps:

- M_0 denote the null model, no predictor, the model simply predicts sample mean
- For $k = 1, ..., p$: Fit all $\binom{p}{k}$ models that contain exactly k predictors
  pick the best among these models. call it M_k
- select the best model among them using cross-vadiated prediction error

2.1.12 Forward/Backward-stepwise selection

Forward stepwise selection The algorithm:

- Let M_0 denote the null model, containing no prectors, predicts sample mean
- For k = 0, ..., p-1
  - consider all p-k models that augument the predictors in M_k with k additional predicot
  - choose the best among these p-k models, call it M_{k+1}.
- select the best model

Backward stepwise selection the algorithm: every step there is one less predictor
Hybird Approach:

- both forward and backward
- vriables are added to model, after each new variable, the method may also remove variables

- attempts to mimic subset selection while retaining the computational advantages of forward and backword stepwise selction

Stepwise methods are more constranit than best subset selection, hence they have lower vairance but perhaps more bias
Shrinkage/regularization methods are computationally efficient.

- LS: $min_\beta(RSS)$
- Shrinkage"$min_\beta(RSS) + \lambda * Pen(\beta)$
  Penalizing

2.2 Regularized regression models

2.2.1 Weighted Least Squares

if we want to give more(or less) weights to different observation if there is Outliers and Heteroscedasticity
Heteroscedasticity (constant variance): $y = X\beta + \epsilon$
$\epsilon_1, ..., \epsilon_n \backsim^{iid} N(0, \sigma^2)$

$$WRSS(\beta) = \sum_{i=1}^{n} w_i \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

$$\hat{\beta_{WLS}} = argmin_\beta(WRSS(\beta)) = (X^T W X)^{-1} X^T W y$$

derivation, see slides

2.2.2 choice of W

In a word it's inversly proportional to variance of x_i

- has to do with variance of y (we have to know the variance(for some measuring device i.e.))
- if repeated measurements of Y for each X is available, then $Var(Y_i|X_i) = Var(\epsilon_i)$ can be estimated
- if we can assume distribution. The variance of proprtions we find should be inversly proportional to the sample size
  so a natrual choice of weights is proportional to the sample size

  Choice of W:

  - assumption: Y_i:average of n_i repeated measurements.
    $w_i = kn_i$

2.2.3 Application of weighted least sqaures

- Forcused accuracy: assign high weights to some observer
- Increased Precision: If we know $Var(\epsilon_i) = \sigma_i^2$, then setting $w = 1/\sigma_i^2$ results in heteroscedastic MLE, i.e. $\hat{\beta}_{WLS} = \hat{\beta}_{ML}$
  $\epsilon_1, ..., \epsilon_n \sim N(0, \sigma_i^2)$ give such weight transfer ϵ to $N(0, 1)$
- Correlated Noise
  we usually assume $Var(\epsilon|X)$ is diagonal matrix or identity matrix.
  but there might be off-diagonal element let $\sum$ be the var-covariance matrix. We know $\sum$ is square, symmetric, positive definite $\rightarrow \sum = BB^T$

we can show that Estimating β in $Y = X\beta + \epsilon$ translates to OLS estimation of β in $B^{-1}y = B^{-1}X\beta + B^{-1}\epsilon$, in this case, $Var(B^{-1}\epsilon) = Var(\epsilon^*) = I$
we can also show $\hat{\beta}_{OLS} = (X^{*T}X)^{-1}X^{*T}y* = (X^T\sum^{-1}X)^{-1}X^T\sum^{-1}y$ this looks like WLS on $y = X\beta + \epsilon$ with weight $w = \sum^{-1}$ with $\sum$ being the variance-covariance matrix

2.2.4 Genralized least squares

WLS problem can be written as miminizing $(y - X\beta)^T W(y - X\beta)$ for a diagonal matrix W
When W is not a diagonal matrix, it is Genralized LS

2.3 Robust regression and breakdown

2.3.1 motivation

- Bump rule
  apply different power transforms to each coordinate
  this removes Outliers
  $$(T_{\alpha x}(X_u), T_{\alpha y})$$
  this doesn't work every tcolorboxenvironment bump rule to straighten scatter plot

- Robust Method
  LS estimate of the parameter is obtained by miminizing the loss function
  $$\sum_{i=1}^{n} r_i^2 = \sum \rho(r_i)$$
  with $\rho(r) = r^2$
  robust regression choose a ρ that make extreme point less significant
  $$S(\beta) = \sum \rho(r_i) = \sum \rho(y_i - x_i^T\beta)$$
  $$\frac{dS(\beta)}{d\beta} = 0 \rightarrow \sum x_i^T \phi(r_i) = 0$$

$\phi(r)$ being $\rho^{'}(r)$, ρ being the loss function
we can let ρ be gentle to outliers to make it robust
closed form means we can isolate β, otherwise it's not closed form

Definition 2.3.1 (M-estimator)
The estimator $\hat{\beta}$ which minimize the function $\sum \rho(r_i)$ is called an M-estimator which is maxinum likelihood type estimator

2.3.2 Iteratively Reweighted Least Squares

it is one way of solving $0 = \sum \phi(r)x$
we can show that
$$0 = \sum \phi(r)x = \sum \frac{\phi(r_i)}{r_i}(y_i - x_i^T\beta)x_i = \sum w(y - x\beta)x = argmin_\beta \sum w(y - x\beta)^2$$
this is same format as weighted least squares $w_i = \phi(r_i)/r_i$. $W = diag(w_1...w_n)$ $\hat{\beta} = (x^TWX)^{-1}X^Ty$
problem is that the weight depends on β, and $beta$ depends on w_i
so we give β an initial value and calculate weight Iteratively untill convergent the algorithm

- set initial value for β
- compute Residuals
- update the weight $w = \frac{\phi(r)}{r}$, W is $diag(w_1...w_n)$
- calculate $\beta^{j+1} = (X^T W^j X)^{-1} X^T W^j y$
- j++, return to step 1 converge if $\beta^{j+1} - \beta^j < \epsilon$

Huber Loss: $\rho(r) = 0.5r^2$ if $|r| \leq c$, or $c(|r| - 0.5c)$ if $|r| > c$ it's quadraitc before c, linear after c
choice of c: if variance
A ϕ function:

- $\phi(-x) = -\phi(x)$
- slope is 1 at 0
- $\phi(x) \geq 0$ for $x \geq 0$; $\phi(x) > 0$ for $0 < x < x_r$
- it then follows from 1 that $\phi(0) = 0$

In practice, we typically need to scale the residuals, $r_i^* = r_i/S$ S is a scaling parameter one choice of S is $MAD = median|r_i|$ and we use $S = MAD/0.6745$ Robust means resistance to outliers

2.3.3 Sensitivity Curve

$$SC(y) = \frac{T_N(y_1...y_{N-1}, y) - T_{N-1}(y_1...y_{N-1})}{1/N}$$

we have property: If the SC of function f is the same as ϕ, the derivative of ρ . Then we have f(x) minimize the ρ

2.3.4 breakdown point

Definition 2.3.2

- Z_m be replacing m of z_i with random desired number
- $e(m, T, Z) = sup_{Z_m^*}|T(Z_m^*) - T(Z)|$
- breakdown point is $min\{m/n : e(m, T, Z) = \infty\}$

we want breakdown point to be as large as possible
we can change LS to least trimmed sqaures

$$\hat{\beta}_{LTS} = argmin_\beta(TrimAverage(y_i - x_i^T\beta)^2)$$

this way the break down point = $\frac{n-k+1}{n}$

3 Locally adaptive methods(smooth functions)

3.1 Local linear regression

3.1.1 Introduction to local regression

$$y = \begin{cases} \beta_1 x \ if \ x \leq a \\ \beta_2 x \ if \ x \geq a \end{cases}$$

subject to $\beta_1 a = \beta_2 a$
we can have continuity condition or no continuity condition

$$y = \beta_0 + \beta_1 x + \beta_2 (x - a) I(x \geq a)$$

we can have differentiability condition:

$$\alpha_1 + 2\alpha_2 a = y_1 + 2y_2 a$$

quadraitc piecewise model can be written as

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 (x - a)^2 I(x \geq a)$$

this model is continuous differentiabile

Definition 3.1.1 (smooth)
twice differentiabile

knots=change-point
some issue:

- the location of the intersecting point (knot) may not be known
- how many knot
- linear model assumption such as constant variance may not hold (completely different variance before and after a)
- we can solve outlier and suddenly changing variance by piecewise model
- the effect of outliers on fitting can be substituted

3.2 model for subset of of data

create subsets of points close to each other

- natrual way of avoiding strong parametric model
- for Example fit only simple linear model
- removes influnce of Outliers

defining a neighbourhood:
For scaler explanatory variate $x \in R$
neighbourhood of x is $\{x_i" | x_i - x| \leq \delta, \forall i = 1...n\}$
or the $||x - x_i||$ distance

k neighbourhood: fixed neighbourhood size, find k nearest neighbours
or $k\%$ nearest neighbours package FNN
KNN local regression:

- gather a fraction $s = k/n$ training points whose x_i are closest to x_0
- assign weight $K_{i0} = 1$ to point in this neighbourhood, zero weight elsewhere

- $$\hat{beta} = argmin \sum K_{i0} \rho(y_i - x_i^T \beta)$$
- the fitted value at $x = x_0$ is $\hat{f}(x_0) = x_0^T \hat{\beta}$

Genralization:

- replacing 0-1 with weight function
- choose robust loss functions for $\rho(r)$ can improve the fit

summary:

- local fitting is only fitting the data locally.
- remove influnce of faraway points
- can have robust local methods (using robustρ)

3.3 Smoothing splines

3.3.1 Geometry of Polynomial Regression

Before we have $\mu(x)\beta_0 + \beta_1 x + \beta_2 x^2$...
we can also have $\mu(x) = \theta_1 g_1(x) + \theta_2 g_2(x)$...

- all linear combinations of these $g(x)$ form a subspace
- g_i are generators of that subspace
- If the generators are lienarly independent of one another
  Then these functions also form a set of orthogonal basis function for that subspace
- The model asserts $\mu(X)$ lies in that subspace, subspace dimention equals the number of basis functions which defines it

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$
$$y = \delta_0 + \delta_1 (x_1 - x_2) + \delta_2 (x_1 + x_2)$$

the latter one is using orthogonal function. the basis the parameter are different

3.3.2 Linear Basis Expansion

- Piecewise constant regression under N_δ and KNN
- Piecewise linear regression under N_δ and KNN
- continuity restriction

$$Y|X = \mu(X) + \epsilon = \sum \beta_m h_m(X) + \epsilon$$

Question: how to choose the basis Answer:

- choose them manually beforehand to limit the class of functions

$$\mu(X) = \sum \mu_j(X_j) = \sum \sum \beta_{jm} h_{jm}(X_j)$$

- include all and use a varialbe selection procedure
- include all and regularize

$$min_\beta [\sum \rho(r_i) + \lambda \sum \beta_j^2]$$

3.3.3 degree of freedom

b-spline is more stable
number of constranit = k continuity + k derivatrive + k 2nd derivative
number of parameter = (k+1)*(degree)
df=k+4

3.3.4 Smoothing spline

decide on number and location of the knots
we define a roughness penalty

$$RSS(\mu, \lambda) = \sum_{i=1}^{n}(y_i - \mu(x_i))^2 + \int (\mu^{''}(t))^2 dt$$

Theorem 3.3.1
suppose f(x) is a real function whose value is known only at $x_1 ... x_n$. The points $(x_1, f(x_1))$ can be ussed to determine natrual cubic splines s(x) s.t. $s(x_i) = f(x_i)$ it can be shown that

$$\int (s^{''}(x))^2 dx \leq \int (g^{''}(x))^2 dx$$

we want to find

$$min_{\mu} \sum_{i=1}^{n}(y_i - \mu(x_i))^2 + \int (\mu^{''}(t))^2 dt$$

the solution requires $\hat{\mu}(x)$ to be $\hat{\mu}(x) = \sum N_j(x)\hat{\beta}_j$, wehre $N_j(x)$ are a set of n basis functions for the family of natrual cubic splines
now we try to find μ, the integer is rewritten as

$$\int (\sum_{j=1} \beta_j N_j^{''}(x))^2 dx = \beta^T \omega_N \beta$$

$\omega = [W_{ij}]$ an n*n matrix, wehre $W_{ij} = \sum N_i^{''}(x) N_j^{''}(x) dx$
$\hat{\mu} = N\hat{\beta} = N(N^T + \lambda\Omega_N)^{-1} N^T y = S_\lambda y$

3.3.5 choosing λ

$\hat{\mu} = N\hat{\beta} = N(N^T + \lambda\Omega_N)^{-1} N^T y = S_\lambda y$

$$df_\lambda = trace(S_\lambda) = \sum \{S_\lambda\}$$

- LOOCV

$$RSS_{CV}(\lambda) = \sum \left(\frac{y_i - \hat{\mu}_\lambda(x_i)}{1 - \{S_\lambda\}_{ii}} \right)^2$$

- Generalized CV

$$RSS_{GCV}(\lambda) = \sum \left(\frac{y_i - \hat{\mu}_\lambda(x_i)}{1 - \frac{1}{n} trace(S_\lambda)} \right)^2$$

3.3.6 Eigen Decomposition

$\rho_1 \geq \rho_2 ... \geq 0$ be eigen values corresponding to eigenvector $u_1, ...u_n$

$$H = \sum u_i \rho_i u_i^T y = \sum u_i \rho_i < u_i, y >$$

For smooth-splines:

$$\hat{\mu} = S_\lambda y \ where \ S_\lambda = N(N^T N + \lambda \Omega_N)^{-1} N^T$$
$$S_\lambda = (I_n + \lambda K)^{-1} \ where \ K = N^{-T} \Omega_N N^{-1}$$
$$S_\lambda = (I_n + \lambda K)^{-1} = (I_n + \lambda V D V^T)^{-1} = V(I_n + \lambda D)^{-1} V^T$$

this is great then eigenvalues is

$$\rho_i(\lambda) = \frac{1}{1 + \lambda d_{n-i+1}}$$

and $\hat{\mu} = \sum \rho_i(\lambda) v_i < v_i, y >$

3.3.7 Multidimensional Splines

how to git a curve to multiple variates? here are several ways:

- Using tensor product basis
- Using a multivariate high curvature penalty: thin plate splines
- Using an additive model

3.3.8 Tensor Product

$$g_{jk}(x_1, x_2) = h_{1j}(x_1) h_{2k}(x_2), j = 1...M_1, k = 1, ...M_2$$
$$\mu(x_1, x_2) = \beta_0 + \sum_j \sum_k \beta_{jk} g_{jk}(x_1, x_2)$$

3.3.9 thin plate splines

solve

$$min\{\sum (y_i - \mu(x_i))^2 + \lambda J[\mu]\}$$

$J[\mu]$ is approperiate penalty function in R^d ($x_i \in R^d$)

$$\mu(x) = \beta_0 + \beta^T x + \sum \alpha_j h_j(x)$$

where

3.3.10 additive spline model

$$\mu(x) = \beta_0 + \mu_1(x_1) ... + \mu_d(x_d)$$
$$J[\mu] = \sum_{j=1} \int \mu_j''(t_j) dt_j$$

3.3.11 Moving beyond linearity

3.4 Kernel method

3.4.1 Local weighting

we choose weight function K(t) s.t.

- $\int K(t)dt = 1$
- $\int tK(t)dt = 0$
- $\int t^2K(t)dt < \infty$

we evaluate kernel function at $\frac{x_i - x}{h}$ x is the location

$$w(x, x_i) = \frac{K(\frac{x_i - x}{h})}{\sum K(\frac{x_j - x}{h})}$$

3.4.2 smoothing matrix svd

$$\hat{\mu} = UD_\rho V^T y = \sum_{i=1}^{n} U_i \rho_i < V_i, y >$$

this separates into basis vectors U_i, singular values ρ_i and rothogonal component of y along direction vectors V_i

- coefficients of y: higher if x is closer to value of x_i
- singular values: have elbow shape, where singular values die off quickly.
- y compononents: $< V_i, y >$:similar pattern to singlar values
- basic functions: increase in complexity as i increase and higher frequency basis functions have small singular values and y components

3.5 Density/intensity function estimate

3.5.1 Lasso and Ridge

$$\hat{\beta}^{lasso} = argmin_\beta \{\sum_{i=1}^{n} (y_i - \beta_0 - \sum x_{ij}\beta_j)^2 + \lambda \sum |\beta_j|\}$$

$$\hat{\beta}^{ridge} = argmin_\beta \{\sum_{i=1}^{n} (y_i - \beta_0 - \sum x_{ij}\beta_j)^2 + \lambda \sum (\beta_j)^2\}$$

3.6 kernel density estimation

$$\hat{f_U}(u) \approxeq \frac{numofu_i in(u-h, u+h)}{2nh}$$

$$\hat{f_U}(u) \approxeq \frac{1}{nh} \sum_{i}^{n} K(\frac{u - u_i}{h})$$

To choose h we use:

- pesudo-likelihood

$$PL(h) = \prod_{i}^{n} \hat{f}_{-i}(u_i)$$

- MISE mean intergrated squared error

$$MISE(h) = E(ISE(h)) = \int E(\hat{f}(x) - f(x))^2 dx = \int \{Var(\hat{f}(x) + Bias^2 \hat{f}(x))\} dx$$

4 Preditive accuracy

4.1 Roles of training and test data, cross-validation, etc

4.2 Bias/Variance trade-off and parameter choice

5 Locally adaptive methods(tree-based methods)

5.1 regression trees