ST3248/EC4308
Lingjie, November 14, 2021

Key problem: estimating the functional relationship in data

Definitions

Supervised	:= Predict an outcome variable Y based on one or more inputs.
Unsupervised	:= No supervising outputs, learns rs between variables.
Error	:= $Y - \hat{Y}$
Bias	:= $E(\hat{\theta} - \theta)$, inflexible model has higher bias
Variance	:= highly flexible method follow data closely
Bias-variance trade-off	:= U shaped test MSE curve, higher flexibility result in low bias but high var
Curse of dimensionality	:= poor model performance at higher dimension
Parametric model	:= model with pre-determined specification
Non-parametric model	:= model without any pre-determined specification
Over-fitting	:= model is tailored to training data and perform badly on test data
Under-fitting	:= model failed to capture the underlying relationship in data
Reducible error	:= error can be reduced with better method or more data
Irreducible error	:= random error ϵ unable to be predict from X

Comment on parametric and non-parametric methods

- Parametric approach will work best when true f is similar to parametric form chosen

  (non-parametric can never beat parametric in this case)
- However, if model specification is wrong, non-parametric outperform parametric in most cases
- Some methods (e.g. KNN) perform worse than parametric methods in high dimension (curse of dimensionality)

Regression problems

key problem: estimate

$$E(Y|X=x) = E(f(x)+\epsilon|X=x) = f(x)$$

Irreducible error, Bias Var trade-off

Mean Squared Error
= reducible + irreducible error
= $bias^2 + var$ + irreducible error

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

where $Y = f(x)+\epsilon$

Classification problems

key problem: estimate

$$p_j(x) = P(Y=j|X=x_0)$$

Irreducible error

Bayes error rate

$$1 - E(\max_j P(Y=j|X))$$

lowest possible test error rate (irreducible error)

Bayes classifier

choose $\hat{y}_0 = j$ from data s.t.

$$\max_j P(Y=j|X=x_0)$$

Bayes decision boundary: $P(Y=1|X) = P(Y=2|X) = 0.5$
Bayes classifier achieves Bayes error rate. However, we need to estimate $P(Y=j|X=x_0)$ before using Bayes classifier.

Model selection

Goal: reduce test error
Problem: training loss underestimate the actual error
Method:

1. Mathematical adjustment (Cp statistics, AIC, BIC)
2. Validation-set approach
3. Cross validation
4. Bootstrap

Regression loss

Mean squared error

$$L(e) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$
$$R(e) = E((y_0 - \hat{f}(x_0))^2)$$

Residual standard error

$$RSE = \sqrt{\frac{1}{n-2}RSS}$$
$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

R^2

$$R^2 = \frac{TSS-RSS}{TSS} = 1 - \frac{RSS}{TSS}$$
$$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

Adj R^2

$$AdjR^2 = 1 - \frac{RSS/(n-d-1)}{TSS(n-1)}$$

Information criterion

Smaller is better
C_p estimates test MSE

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$$

AIC (Akaike IC) for prediction

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

BIC (Bayesian IC) for true model, prefer smaller model

$$BIC = \frac{1}{n\hat{\sigma}^2}(RSS + \log(n)d\hat{\sigma}^2)$$

Note

- require to know number of features in considered model d
- outcome variable y and n must be the same
- require estimation of the response variance $\hat{\sigma}^2$ (can be estimated with model with all predictors)
- $\log n > 2$ if $n > 7$, BIC penalize training more heavily than AIC when $n > 7$

Classification loss

Error rate

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$
$$R(y,\hat{y}) = E(I(y_0 \neq \hat{y}_0)) = P(y_0 \neq \hat{y}_0 | X_0)$$

Deviance

Generalization of "least squares" for general linear models. Measures distance between data and fit. Min = better

$$Deviance = -2log(Likelihood) + Constant \geq 0$$
$$\sim \chi^2_{df=Residual}$$

In linear models, deviance is proportional to RSS

Confusion matrix

True class (horizontal)

		Positive	Negative
Predicted class	Positive	TP	FP
	Negative	FN	TN

False positive (FP) rate	:=	% negative examples that are classified as positive
False negative (FN) rate	:=	% positive examples that are classified as negative
Sensitivity	:=	True positive rate $(TP/(TP+FP))$
Specificity	:=	True negative rate $(TN/(TN+FP))$

AUC, ROC

ROC: receiver operating characteristic curve
AUC: area under the curve

- Compares FP rate (x-axis) and TP rate (y-axis)
- Higher AUC, larger area = better
- Useful to compare different probability threshold
- When threshold = 1, FP=0, TP=0 (lower left)
- When threshold = 0, FP=1, TP=1 (upper right)

Note: $\hat{P}(Y|X) \geq$ threshold will be classify as positive

Cross-validation (CV)

One standard error rule: choose simplest model with test error estimate within one standard error of the min value

Validation-set approach

Divide samples into:
1. Training set 2. Validation/hold-out set

```
1. For $i = 1, \cdots, N$:
      [1.1] Randomly split $n$ samples into train, test group
      [1.2] For model $j = 1, \cdots, M$:
         [1.2.1] Fit model $j$ on training group $i$
         [1.2.2] Evaluate model $j$ based on test group $i$
2. Plot $i$th performance (e.g. MSE) against model $j$
```

Limitation

- High variance
- Overestimate test error as only subset of data is used in training (bias is reduced with more data possible)

K-fold Cross-validation

```
1. Randomly split $n$ samples into $K$ blocks, each block has $n_K = n/K$ obs
2. For model $j = 1, \cdots, M$:
      [2.1] For block $k = 1, \cdots, K$
         [2.1.1] Fit model $j$ on all blocks except block $k$
         [2.1.2] Evaluate model $j$ based on block $k$
      [2.2.2] Calculate $CV^{(j)}_{(K)} = \sum_{k=1}^{K}(n_k/n)MSE_k$
      Note $(n_k/n) \approx 1/K$
```

Estimated test error

$$CV_{(K)} = \frac{1}{K}\sum_{i=1}^{K} MSE_i$$

Variance

$$\hat{Var}(CV_{(K)}) = \sum_{k=1}^{K}\frac{(MSE_k - \bar{MSE_k})^2}{K-1}$$
$$= \frac{1}{K}\hat{Var}(CV_k) + 2\sum_{i<j}^{K} Cov(CV_i, CV_j)$$

Limitation

- Bias still exist (min bias when $K = n$)
- CV should be performed before feature selection step

Leave-one out CV (LOOCV)

Using Cross-validation with $K = n$
Special result for least-squares linear, polynomial regression

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2$$

$h_i :=$ leverage, $n :=$ number of samples (i.e. LOOCV)
Limitation

- High variance since estimates from each fold might be highly correlated
- Computational intensive

Bootstrap

Useful to quantify uncertainty associated with a given estimator or statistical learning method. Usually used to quantify standard error of an estimator and prediction error (sd)
Key idea: resampling with replacement

1. Randomly sample n $(Z^{*(r)}, r = 1, \cdots, B)$ obs from n samples with replacement
2. Estimate θ_i with $Z^{*(r)}$
3. $\hat{\theta} = \frac{1}{B}\sum_{r=1}^{B}\hat{\theta}^{*(r)}$

$$SE_B(\hat{\theta}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\theta}^{*(r)} - \hat{\theta}^{*})^2}$$

Bootstrap for time series

Create blocks of consecutive observation and sample with replacement

Preprocessing

Standardization of Numeric Features

Min-Max normalization

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Z-score standardization

$$X' = \frac{X - \mu}{\sigma}$$

Standardizing the predictors

$$X' = \frac{X}{\sigma}$$

Transforming Nominal Features

One hot encoding

$$X' = \begin{cases} 1, x = \text{class } 1 \\ 0, x = \text{class } 2 \end{cases}$$

Classic Regression models

K Nearest Neighbourhood regressor

Estimate the mean outcome with the K nearest obs

$$f(x) = E(Y|X \in N(x))$$
$$\hat{f}(x) = \frac{1}{K} \sum_{i \in N(x)} y_i$$

where $N(x)$ is some neighbourhood of x
$K :=$ number of neighbourhood, optimal choice depends on bias-variance trade-off
$1/K :=$ flexibility, higher (small K) is more flexible (low bias) but overfitting (high variance)

Linear Regression

$$\begin{aligned} y_i = \beta_0 + \beta_1 x_1 + \\ + \beta_2 x_1 x_2 + \beta_3 I(x_3 = 1) + \beta_4 x_1^2 \\ + \epsilon_i \end{aligned}$$

qualitative var	: $I(x_3 = 1)$
interaction effect	: $x_1 x_2$
non-linear	: x_1^2

Note:

- Due to hierarchical principle, should always include the main effects if the interaction is significant

Answering the questions

1. Is there a relationship between X and y

   [a] Test $F - stat$ hypothesis

   $$H_0 : \beta = 0, H_1 : \beta \neq 0$$

   [b] Visual plot X, y

   [Conclude] there is/no linear/non-linear relationship

2. How strong is the relationship?

   RSE, R^2

3. Which X contribute to y

   Examine p-value for $t - stat$ (individual) or $F - stat$ (categorical var)

4. How large is the effect of each X on y?

   Confidence interval for β

   Collinearity: VIF

5. How accurately can we predict future y?

   Report $\hat{Y}$

   For $E(\hat{Y})$, use CI

   For $\hat{Y}$, use PI (capture additional uncertainty of irreducible error)

6. Is the relationship linear

   Error plot to ensure no trend in errors

7. Is there interaction effect between X?

   Fit interaction terms and test for significant and R^2

Estimate coefficients

Least squares estimates

$$\begin{aligned} \min_{\beta_0, \beta_1} RSS &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x))^2 \\ \hat{\beta}_1^* &= \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \\ \hat{\beta}_0^* &= \bar{y} - \hat{\beta}_1^* \bar{x} \end{aligned}$$

Note:

- Least squares estimator is unbiased

Standard error

Standard error = standard deviation of the estimator

$$\begin{aligned} \epsilon_i &\sim (0, \sigma^2) \\ SE(\hat{\beta}_0)^2 &= \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \right] \\ SE(\hat{\beta}_1)^2 &= \frac{\sigma^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \end{aligned}$$

Note:

- Require data to be closer together $\Rightarrow$ low σ^2
- Require data to have a wider range $\Rightarrow$ high $Var(X)$

Residual standard error

Estimate σ^2, variance of ϵ_i

$$\hat{\sigma} = RSE = \sqrt{\frac{RSS}{n-2}}$$

Confidence intervals

$\Phi(\alpha/2) = F^{-1}(\alpha/2)$, quantile of standard normal

$$\hat{\beta}_j \pm \Phi(\alpha/2) \cdot SE(\hat{\beta}_j)$$

Hypothesis tests

Testing

$$H_0 : \beta_i = 0, \ H_1 : \beta_i \neq 0$$
$$t_{obs} = \frac{\hat{\beta}_i - 0}{SE(\hat{\beta}_i)} \sim t_{n-p-1}$$

$$\begin{aligned} &H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0 \\ &H_1 : \beta_i \neq 0, \text{ for any } i \\ &F_{obs} = \frac{(TSS - RSS)/p}{RSS/(n-p-1)} \sim F_{p,n-p-1} \end{aligned}$$

$$\begin{aligned} &H_0 : \beta_{p-q+1} = \cdots = \beta_p = 0 \\ &H_1 : \text{ Not } H_0 \\ &F_{obs} = \frac{RSS_0 - RSS/q}{RSS/(n-p-1)} \sim F_{p,n-p-1} \end{aligned}$$

Note:

- $p :=$ number of regressors, excluding β_0
- either assume $\epsilon \sim N(0, \sigma^2)$, else by by LLN test statistics still holds when n is large
- TSS = total sum of squares
- RSS = residual sum of squares
- F statistics is required as there is α% chance of not rejecting t-stat even though the model is insignificant

Deciding on important variables

- Forward selection
- Backward selection
- Mixed selection

Prediction

Sources of uncertainty

- Model bias: $\hat{f}(X) - f(X)|X$
- Irreducible error: ϵ

Two intervals

- Confidence interval: only quantify estimation uncertainty
- Prediction interval: quantify both estimation uncertainty and irreducible.

  Wider than CI

  Require normality assumption for ϵ

Inference/Interpretation

simple model	:	β_j is the average change in Y in the model when X_j is changed by one unit holding all other predictors fixed
qualitative	:	β_j is the difference between group 1 and 2
	:	β_0 is the average effect for base group

Multicollinearity

Collinearity refers to situation when two or more predictors are closely related to one another.
Issues

- Hard to determine individual impact of collinear predictors on response
- Reduce accuracy of the regression coefficient estimate and increase standard error of β_j, decline in the true T-stat
- May cause β_j and x_j to relate to response when they are not. Reduce effectiveness of null hypothesis

VIF Test:

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{xj|x-j}}$$

where $R^2_{xj|x-j}$ is the R^2 from a regression of X_j onto all of the other predictors

Error analysis

- Non-linearity of the response-predictor relationship

  Observe residual plot ($e_i = y_i - \hat{y}_i$) to ensure "patternless" - without trend

- Correlation of the error terms

  $E(\epsilon_i|\epsilon_{i-1}) = 0$

  correlated error term result in underestimated standard error - overconfident in inferences

  Observe residual over time

- Non-constant variance of error terms

  $Var(\epsilon_i) \neq \sigma^2$

  Observe for non "funnel shape" residual plot

  Consider $Y' = log(Y), \sqrt(Y)$

- Outliers

  Outliers: values that don't fit the pattern on rest of data (comparing $Y, \hat{Y}$)

  Does not affect least squares fit, but large effect on RSE - therefore standard error and intervals

  Observe studentized residual - magnitude more than 3 is considered outlier

- High-leverage points

  Unusual in terms of features (only compare X, x_i)

  require calculation of leverage in > 2 dimension

- Influential point

  Both large studentized residual (outlier) and large leverage

- Collinearity

  Several predictors are strongly linearly related

  Affect standard errors - intervals

  require calculation of measures such as VIF in > 2 dimension

Extending linear regression

Subset selection

- Identify irrelevant features and set their coefficients exactly to zero
- Best subset selection perform best in-sample, out-of-sample performance is uncertain
- Best subset, forward and backward might have different models
- model K in Forward and Backward selection is a subset of model $K + 1$

Best subset selection

1. Let M_0 denote intercept model (predicts sample mean)
2. For $k \in [1, p]$

   [a] Fit all $\binom{p}{k}$ models contain exactly k predictors

   [b] Let M_k be the best model among $\binom{p}{k}$ models (selected with smallest RSS, or largest R^2)
3. Select single best model from $M_0, M_1, \cdots, M_p$ using cross-validation, $C_p(AIC), BIC$ or adjusted R^2 (or deviance for classification problems)

Forward stepwise selection

1. Let M_0 denote intercept models
2. For $k = 0, \cdots, p-1$

   [a] Consider all $p - k$ models that adds one new variable to M_k

   [b] Denote M_{k+1} as the best model among those $p - k$ models
3. Select single best model from $M_0, \cdots, M_p$

Backward stepwise selection

1. Let M_0 denote intercept models
2. For $k = p, p-1, \cdots, 1$

   [a] Consider all k models that contain all but one predictors in M_k for a total of $k - 1$ predictors

   [b] Denote M_{k+1} as the best model among those k models
3. Select single best model from $M_0, \cdots, M_p$

Shrinkage methods

- Use all p features in model but shrunk estimates towards 0
- Variable selection when coef are exactly 0, only in Lasso
- Shrinkage methods are biased, but reduce variance
- Shrinkage methods are have lesser degree of freedom than least squares (constrain or regularize coefficient estimates)
- Require scaling before model fitting

Ridge (L2 norm)

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

ℓ_2 norm

$$||\beta||_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

Note:

- Plotting coefficients versus λ with

$$||\hat{\beta}_\lambda^R||_2 / ||\hat{\beta}||_2$$

  when $\lambda = 0 \Rightarrow$ term $= 1$

  when $\lambda \to +\infty \Rightarrow$ term $=1$

- When express as constrained optimization problem, larger budget $s \Leftrightarrow$ small λ

  small budget $s \Rightarrow$ coefficients shrink towards 0

Lasso (L1 norm)

yields sparse, easier to interpret models

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

ℓ_1 norm

$$||\beta||_1 = \sum_{j=1}^{n} |\beta_j|$$

Special case of Ridge and Lasso

Conditions

- $n = p$
- $\boldsymbol{X}$ is identity matrix
- no intercept model $Y_j = \beta_j + \epsilon_j$

OLS estimate: $\beta_j = y_j$
Special result for ridge

$$\hat{\beta}_j^R = \frac{y_j}{1+\lambda}$$

Special result for lasso

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2, & y_i > \lambda/2 \\ y_j + \lambda/2, & y_i < -\lambda/2 \\ 0, & |y_j| \le \lambda/2 \end{cases}$$

Result

- ridge shrinks by multiplicative factor [continuous]
- lasso shrinks by constant $\lambda/2$, but thresholds to 0 when $|y_j| \le \lambda/2$ (soft thresholding) [break point]

Dimension reduction methods

Objective: fit least squares on a projected variable with M dimension, where $M < p$, the original dimension
Let $Z_1, \cdots, Z_m$ denote projected variables

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

We have constants $\phi_{1m}, \cdots, \phi_{pm}$ which forms projected variables as linear combination of original features

Least square on projected features

Fit linear regression on projected variables

$$y_i = \phi_0 + \sum_{m=1}^{M} \phi_m z_{im} + \epsilon_i, i \in [1, n]$$

Can be viewed as constraining coefficients

$$\sum_{m=1}^{M} \phi_m z_{im} = \sum_{m=1}^{M} \theta_m \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)$$

$$= \sum_{j=1}^{p} \left(\sum_{m=1}^{M} \theta_m \phi_{jm} \right) x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij}$$

Principal component analysis

PCA reduce features with minimal loss of information by maximising variance of the projected variables
ϕ_{m1} := principal components loading, $\sum_{m=1}^{M} \phi_{m1}^2 = 1$
z_{i1} := principal component scores, $i \in [1, n]$
Note:

- first principal component direction of the data is the direction along which observation vary the most
- first principal component vector defines the line that is as long as possible to the data (orthogonal)
- with p variables, max p principal components
- key assumption: the direction in which features $X_1, \cdots, X_p$ are most variable are the directions that are associated with Y
- Principal components regression (PCR) does not do variable selection, each principal component is a non-sparse linear combination of the original feature
- PCA should be done after standardizing predictors, unless predictors are already in the same scale

Partial least squares

PLS learns directions in supervised way

- Require standardizing predictors
- Z_1 is computed by setting ϕ_{j1} (weight on X_j in the projection) to be the simple linear regression coefficient when regressing Y on X_j
- PLS places most weight on features that are highly correlated to the response
- PLS explains more variation of X with Y, PCA explains more variation in X alone

Classic Classifiers

K Nearest Neighbourhood classifier

Estimate class prob by fraction of class j in K nearest obs

$$P(Y = j | X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

K := number of neighbourhood
$1/K$:= flexibility, higher is more flexible but overfitting

Logistic regression

$$p(X) = P(Y=1|X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$
$$\Rightarrow \log\left(\frac{p(X)}{(1-p(X))}\right) = \beta_0 + \beta_1 X$$
$$P(Y=k|X) = \frac{exp(X\beta)}{\sum_{i=1}^{K} exp(X\beta)}$$

- $\log(\cdot)$ is called log odds or logit transformation of $p(X)$
- Log odds can take any real value, and we model linear in log odds
- β_1 represents the increase in the log odds of $Y=1$ for a one unit increase of X
- No interpretation between β_1 and $p(X)$ for a one-unit increase in X
- Regression estimates gives z-statistics by LLN (approximate distribution instead of exact t-stat in linear regression)

Linear probability regression

- For binary outcome, linear regression = linear discriminant analysis
- Coding response Y as an integer implies an ordering
- However, linear regression produce probability outside $(0,1)$
- Linear regression does not work for more than 2 classes

Estimation: MLE and Deviance

Assume Bernoulli distribution

$$f_i(y_0) = p(x_i)^{y_i}(1-p(x_i))^{1-y_i}, i \in [1,n]$$
$$\begin{aligned} lik(\beta_0,\beta) &= \prod_{i=1}^{n} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} \\ &= \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1-p(x_i)) \\ &= \prod_{i=1}^{n} \left(\frac{exp(x_i'\beta)}{1+exp(x_i'\beta)}\right)^{y_i} \left(\frac{1}{1+exp(x_i'\beta)}\right)^{1-y_i} \end{aligned}$$
$$\ell(\beta_0,\beta) = \sum_{i=1}^{N} \left[log(1+e^{(\beta_0+x_i'\beta)}) - y_i(\beta_0+x_i'\beta)\right]$$

$$\begin{aligned} Deviance &\propto \sum_{i=1}^{N} [log(1+e^{\beta_0+x_i'\beta} - y_i(\beta_0+x_i'\beta)] \\ &= -2\ell(\beta_0,\beta) + C \end{aligned}$$

Note:

- C relates to the likelihood of the "perfect" model
- Null deviance D_0: $\ell(\hat\beta_0)$ (intercept only)
  $\ell(\hat\beta_0)$ small $\Rightarrow D_0$ large
- Perfect fit (overfitting): $\ell(\hat\beta_0,\hat\beta_1) = C/2$

Interpretation

odds	$:= p(X)/(1-P(X)) \in (0,\infty)$
log odds	$:= log(p(X)/(1-P(X))) \in (-\infty,\infty)$
β_0	$:=$ log-odds of $Y=1$ when $X=0$
β_1	$:=$ increase in the log odds of $Y=1$ for a one unit increase of X

Unbalanced dataset: Case-control sampling

$$\hat\beta_0^* = \hat\beta_0 + \log(\frac{\pi}{1-\pi}) - \log(\frac{\tilde\pi}{1-\tilde\pi})$$

$\tilde\pi :=$ sample prevalence (proportion)
$\pi :=$ population prevalence (proportion)

- Case-control is when we intentionally oversampled minority class, usually due to class imbalanced.
- β_j will be accurate, but β_0 require adjustment
- Sampling more controls than cases reduces the variance of the parameter estimates
- Diminishing return: ratio of 5:1 (control:case, negative:positive) is best

Discriminant Analysis

Based on Bayes theorem

$$\begin{aligned} P(Y=k|X=x) &= \frac{P(X=x|Y=k)P(Y=K)}{P(X=x)} \\ &= \frac{\pi_k f_k(x)}{\sum_{i=1}^{K}\pi_i f_i(x)} \\ &\propto \pi_k f_k(x) \end{aligned}$$

$f_k(x) :=$ density for X in class k
$\pi_k :=$ marginal/ prior probability for class k
Note:

- We compare $\pi_k f_k(x)$ to determine the highest probability
- When classes are well-separated, LDA perform better than LR
- When assumption is right (normal distribution) and/or n is small, LDA outperform LR
- LDA can be used to provide low-dimensional views of data
- More commonly used for multi-class classification problem than logistic regression
- Benchmark method for classification
- Is a parametric model

Linear Discriminant Analysis (LDA)

Assume $f_k(x) \sim N(\mu_k, \sigma^2)$, normal with same variance

$$\begin{aligned} f_k(x) &= \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2\right) \\ &= \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \end{aligned}$$

Quadratic Discriminant Analysis (QDA)

Assume $f_k(x) \sim N(\mu_k, \sigma_k^2)$, normal with diff variance

$$\begin{aligned} f_k(x) &= \frac{1}{\sqrt{2\pi}\sigma_k} exp\left(-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2\right) \\ &= \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma_k^{-1}(x-\mu)\right) \end{aligned}$$

Discriminant functions

By simplifying and cancelling, we can assign x to the class with largest discriminant score
LDA: linear (in x) discriminant function

$$\begin{aligned} \delta_k(x) &= x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \\ &= x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log(\pi_k) \end{aligned}$$

Decision boundary ($\delta_k(x) = \delta_l(x)$): when
$K=2, \pi_1=\pi_2=0.5 \Rightarrow x = (\mu_1+\mu_2)/2$
Assume
$\pi_k = \pi_l \Rightarrow x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k = x^T\Sigma^{-1}\mu_l - \frac{1}{2}\mu_l^T\Sigma^{-1}\mu_l$
QDA: quadratic (in x) discriminant function

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

Estimating parameter (LDA)

$$\hat{\pi}_k = \frac{n_k}{n}$$
$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i$$
$$\hat{\sigma}^2 = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i-\hat{\mu}_k)^2$$
$$= \sum_{k=1}^{K}\frac{n_k-1}{n-K}\hat{\sigma}_k^2$$

Fisher's Discriminant Plot (LDA)

Use $K-1$ dimensional plot to represent K classes
When $K>3$, find "best" $2-d$ plane for visualisation

From $\delta_k(x)$ to probability

$$\hat{P}(Y=k|X=x) = \frac{\pi_k exp(-\frac{1}{2}\left(\frac{(x-\mu_k)^2}{\sigma^2}\right))}{\sum_{i=1}^{K}\pi_k exp(-\frac{1}{2}\left(\frac{(x-\mu_k)^2}{\sigma^2}\right))}$$

Naive Bayes

Assume independent features

$$f_k(x) = f_{k1}(x_1)f_{k2}(x_2)\cdots f_{kp}(x_p)$$

Laplace estimator

- problem: $P(x_i|k)=0 \Rightarrow P(Y|X)=0$
- solution: add 1 to counts
- Note: denominator increase by p

Estimating parameter (NB)

Estimate quantitative X_j as univariate normal

$$X_j|Y=k \sim N(\mu_{jk},\sigma_{jk}^2)$$

Estimate qualitative X_j as proportion count

$$\hat{f_{kj}}(x_j) = \frac{N_{kj}}{N_j}$$

From NB to probability

$$\hat{P}(Y=k|X=x) = \frac{\pi_k f_k(X_k)}{\sum_{i=1}^{K}\pi_k f_k(X_k)}$$

Comparing classification methods

Logistic regression is relate to LDA

$$log\left(\frac{p_1(x)}{1-p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x$$
$$c_0(\mu_1,\mu_2,\sigma^2), c_1(\mu_1,\mu_2,\sigma^2)$$

difference in estimation method: MLE vs estimating normal class feature distributions

Tree-based methods

key: stratifying, segmenting predictor space into number of simple regions

Terminology

terminal nodes	: ending leaves, or regions in graph
splitting	: $X = \{X_j < t_k, X_j \geq t_k\}$
internal nodes	: t_k, predictor space is split

Decision tree

advantage:

- simple, useful for interpretation
- graphical plotting
- close to human decision-making

disadvantage:

- low predictive power

Tree building

Top down	: start at top of tree then successively splits
Greedy approach	: best split is made at each step, rather than future steps
Prediction	: mean outcome in region R_j

1. Select cutpoint s for X_j such that $\{X|X_j < s\} \cup \{X|X_j \geq s\}$ minimises RSS
2. Repeat (1) for subsequent predictors, based on the previous identified regions
3. Process repeat till stopping criterion

$$\min_{R_j}\sum_{j=1}^{J}\sum_{i\in R_j}(y_i - \hat{y_{R_j}})^2$$

R_j: region j
$\hat{y}_{R_j}$: average outcome within jth box

Tree pruning

Raw tree overfit training data
Solution: smaller tree with lower variance but some bias

S1 : set high threshold RSS split criterion

However, this is short-sighted as split might leads to large reduction in RSS later on

S2 : Cost complexity pruning/ weakest link pruning

Grow a large tree, then prune back to obtain subtree

There exist a subtree $T \subset T_0$ for each α s.t.

$$\min_{\alpha}\sum_{m=1}^{|T|}\sum_{i:x_i\in R_m}(y_i - \hat{y}_{R_m})^2 + \alpha|T|$$

$|T|$: number of terminal nodes
R_m: rectangle corresponding to the mth terminal node
$\hat{y}_{R_m}$: mean outcome in R_m
α: controls trade-off between subtree complexity and training fit, selected through CV before applying to full data

1. Use recursive binary splitting to grow large tree, stop when terminal nodes have few obs
2. Obtain sequence of best sub trees (function of α) with cost complexity pruning
3. Use K-fold CV to select α. For $k=1,\cdots,K$

   Repeat step 1 and 2 on $(K-1)/K$ fraction of data, less k fold

   Evaluate MSE based on k fold, as a function of α

   Average MSE and pick α to min average error
4. Return subtree corresponds to the chosen values of α

Classification tree

$\hat{y}$: most commonly occurring class
$\hat{p}_{mk}$: proportion of obs in m region from k class

Loss function

Classification error rate (not sensitive to tree-growing)

$$E = 1 - \max_k(\hat{p}_{mk})$$

Gini index (measure purity)

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

G is small when $\hat{p}_{mk}$ close to 0 or 1 (same class)
Entropy (info gain)

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

Bagging

Applying bootstrap to generate B samples and take average
$Var(\bar{Z}) = \sigma^2/n$ (i.i.d)

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*(b)}(x)$$

For classification: majority vote

Out-of-Bag Error Estimation

out-of-bag (OOB) observation: obs not used in fitting a given tree in bagging

Predict ith obs using trees where obs is OOB ($\approx B/3$ trees)

If B is large, OOB estimate is essentially LOOCV

Random Forests

RF decorrelates bagged trees, reducing var (non-i.i.d)

key: random selection of m predictors, $m = \sqrt{p}$ (common)

Bagging: $m = p$ (all predictors)

Boosting

Fit small, low variance "weak leaders" sequentially and aggregate them slowly to improve forecasting

Loss function

$$\min_{\{\beta^{(m)},\gamma^{(m)}\}_1^M} \sum_{i=1}^{N} L\left(y_i, f(x; \beta^{(m)}, \gamma^{(m)})\right)$$

with Basis function $f(x)$

$$f(x; \beta^{(m)}, \gamma^{(m)}) = \sum_{i=1}^{M} \beta^{(m)} b(x; \gamma^{(m)})$$

Iterative update version

$$\Rightarrow \min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f^{(m-1)} + \beta b(x_i; \gamma))$$

Hyper-parameter:

- number of trees B, larger is better
- shrinkage parameter λ, small learning rate
- number of splits d, smaller is better

Forward stagewise additive modelling (FSAM)

1. Initialize $f^{(0)}(x) = 0$
2. For $m = 1, \cdots M$:

   [a.] Compute

   $$(\beta^{(m)}, \gamma^{(m)}) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f^{(m-1)}(x_i) + \beta b(x_i, \gamma))$$

   [b.] set $f^{(m)}(x) = f^{(m-1)}(x) + \beta^{(m)} b(x, \gamma^{(m)})$
3. Final $f(x) = f^{(M)}(x) = \sum_{m=1}^{M} \beta^{(m)} b(x, \gamma^{(m)})$

Gradient Boosting

Friedman (1999) estimates gradient with tree model ($h^{(m)}$) and learning rate ($\lambda \leq 1$)

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
2. For $m = 1, \cdots M$:

   [a.] Compute

   $$r_i^{(m)} = -\frac{\partial L(y_i, f^{(m-1)}(x_i))}{\partial f^{(m-1)}(x_i)}, i \in [1, N]$$

   [b.] Fit base learner $h^{(m)}(x)$ to $\{r_i^{(m)}\}_{i=1}^N$

   [a.] Compute step length $\beta^{(m)}$ for $h^{(m)}(x)$

   $$\beta^{(m)} = \arg\min_\beta \sum_{i=1}^{N} L(y_i, f^{(m-1)}(x_i) + \beta h^{(m)}(x_i))$$

   [b.] set $f^{(m)}(x) = f^{(m-1)}(x) + \lambda \beta^{(m)} h^{(m)}(x)$
3. Final $f(x) = f^{(M)}(x) = \sum_{m=1}^{M} \lambda \beta^{(m)} f^{(m)}(x)$

Gradient in gradient boosting

Deriving

$$r_i^{(m)} = -\frac{\partial L(y_i, f^{(m-1)}(x_i))}{\partial f^{(m-1)}(x_i)}, i \in [1, N]$$

Solve

$$\hat{f} = \arg\min_f L(y_i, f(x_i))$$

Arriving at

$$-\frac{\partial L(y_i, f(x))}{\partial f(x)} = -r_i^{(m)}$$

Regression gradient example

$$-\frac{\partial \frac{1}{2}(y_i - f(x_i))^2}{\partial f(x)} = y_i - f(x_i)$$

$$-\frac{\partial |y_i - f(x_i)|}{\partial f(x)} = sign(y_i - f(x_i))$$

$$-Huber = \begin{cases} \frac{1}{2}(y_i - f(x_i)) \\ \delta(|y_i - f(x_i)| - \frac{1}{2}\delta) \end{cases} \Rightarrow$$

$$\begin{cases} y_i - f(x_i), & |y_i - f(x_i)| \leq \delta \\ \delta \cdot sign(y_i - f(x_i)), & |y_i - f(x_i)| > \delta \end{cases}$$

Therefore, the optimal $f^{(0)}(x), r_i^{(m)}, \beta^{(m)}(\gamma^{(m)})$ for MSE loss

$$f^{(0)}(x) = \bar{y}$$

$$r_i^{(m)} = y_i - \hat{y}_i^{(m-1)}$$

$$\beta^{(m)} = \frac{\sum_{x_i \in R_j^{(m)}} r_i^{(m)}}{count(R_j^{(m)})}$$

Implementation (MSE)

1. Initialize $f^{(0)}(x) = \bar{y}$
2. For $m = 1, \cdots, M$:
   [a] For $i \in [1, N]$, compute
$$r_i^{(m)} = y_i - \hat{y}_i^{(m-1)}$$
   [b] Fit regression tree to $r^{(m)}$ with $d+1$ terminal nodes and terminal regions $R_j^{(m)}, j \in [1, J^{(m)} = d+1]$
   [c] For $j \in [1, J^{(m)}]$, compute
$$\gamma_j^{(m)} = \frac{\sum_{x_j \in R_j^{(m)}} r_i^{(m)}}{count(R_j^{(m)})}$$
   [d] Update
$$f^{(m)}(x) = f^{(m-1)}(x) + \lambda \sum_{j=1}^{J^{(m)}} \gamma_j^{(m)} I(x \in R_j^{(m)})$$
3. Output $\hat{f}(x) = f^{(M)}(x)$

Classification gradient example

$$-Deviance \Rightarrow I(y_i = G_k) - p_k(x_i)$$
$$(binary) \begin{cases} 1 - p_i(x_i), & y_i = 1 \\ -p_i(x_i), & y_i = 0 \end{cases}$$

Therefore, the optimal $f^{(0)}(x), r_i^{(m)}, \beta^{(m)}(\gamma^{(m)})$ for Deviance

$$f^{(0)}(x) = \frac{\sum_{i=1}^N y_i}{N}$$
$$r_i^{(m)} = y_i - p_i^{(m)}$$
$$\beta^{(m)} = \frac{\sum_{x_i \in R_{ij}^{(m)}} (y_i - p_i^{(m)})}{\sum_{x_i \in R_{ij}^{(m)}} p_i^{(m)}(1 - p_i^{(m)})}$$
$$= \frac{\sum_{x_i \in R_{ij}^{(m)}} r_i^{(m)}}{\sum_{x_i \in R_{ij}^{(m)}} p_i^{(m)}(1 - p_i^{(m)})}$$

in binary case

$$p_i^{(m)} = \frac{exp(log(odds))}{1 + exp(log(odds))}$$
$$= \frac{exp(f^{(m-1)}(x))}{1 + exp(f^{(m-1)}(x))}$$

Implementation (K-class)

1. Initialize $f_k^{(0)}(x) = 0, k \in [1, K]$
2. For $m = 1, \cdots, M$:
   [a] Set
$$p_k^{(m)}(x) = \frac{exp(f_k^{(m-1)}(x))}{\sum_{j=1}^K exp(f_j^{(m-1)}(x))}, k \in [1, K]$$
   [b] Compute $r_{ik}^{(m)} = y_{ik} - p_k^{(m)}(x_i), i \in [1, N]$
   [c] Fit a tree to $\{r_{ik}^{(m)}\}_{i=1}^N$ with terminal regions $R_{jk}^{(m)}, j \in [1, J^{(m)}]$
   [d] Form $\gamma_{jk}^{(m)}$
$$\gamma_{jk}^{(m)} = \frac{K-1}{K} \frac{\sum_{x_i \in R_{jk}^{(m)}} r_{ik}^{(m)}}{\sum_{x_i \in R_{jk}^{(m)}} |r_{ik}^{(m)}| \cdot (1 - |r_{ik}^{(m)}|)}, j \in [1, J_m]$$
   [e] Update $f_k^{(m)}$
$$f_k^{(m)}(x) = f_k^{(m-1)}(x) + \lambda \sum_{j=1}^{J^{(m)}} \gamma_{jk}^{(m)} I(x \in R_{jk}^{(m)})$$
3. Output $\hat{f}_k(x)$
$$\hat{f}_k(x) = f_k^{(M)}(x), k \in [1, K]$$

Variable importance measure

Ordered by

- (Regression) Amount of RSS reduced due to splits over a given predictor, averaged over all B trees
- (Classification) Amount of Gini index decreased by splits over a given predictor, averaged over all B trees

Support Vector Machine

Expanding SVM for more than 2 classes:

- OVA: One versus All

  Fit K different 2-class SVM classifiers for each class vs the rest. Classify data to class which $\hat{f}_k(x^*)$ is largest
- OVO: One versus One

  Fit all $\binom{K}{2}$ pairwise classifiers and classify x^* to class that wins the most pairwise competitions

use OVO if K is not too large

Maximal Margin Classifier

Constrained quadratic programming

$$\max_{\beta_0, \beta} M$$
$$s.t. \sum_{j=1}^p \beta_j^2 = 1, y_i \left(\beta_0 + \sum_{j=1}^p \beta_j x_{ip}\right) \geq M, i \in [1, N]$$

1. Maximise margin
2. subject to length of normal vector = 1
3. and the distance of a point is larger than margin and is on the right side of the margin (distance: $X^T w + \beta_0 = d$)

Alternative representation

$$\min_{w, \beta_0} \frac{1}{2} ||w||^2$$
$$s.t. \ y_i(x_i^T w + \beta_0) \geq 1, i \in [1, N]$$

1. Minimise $\frac{1}{M} = ||w||, w = (\beta_1, \beta_2, \cdots, \beta_p)^T$
2. Equivalent to original programming by fixing $M = 1$

Lagrange primal problem

$$\min_{w, \beta_0} L_p = \frac{1}{2} ||w||^2 - \sum_{i=1}^N \alpha_i [y_i(x_i^T w + \beta_0) - 1]$$

1. by KKT, support vectors are points with $\alpha_i > 0$ (active constraint)
$$\alpha_i = \begin{cases} 0, & y_i(w_i^T w + \beta_0) - 1 > 0 \\ > 0, & y_i(w_i^T w + \beta_0) - 1 = 0 \end{cases}$$
2. Solutions to the primal problem
$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^N \alpha_i y_i x_i$$
$$\frac{\partial L}{\partial \beta_0} = 0 \Rightarrow 0 = \sum_{i=1}^N \alpha_i y_i = y^T \alpha$$
$$\text{Dual:} \ \Rightarrow L = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i^T x_j)$$

Note: $\alpha_i \alpha_j y_i y_j$ is scalar, $x_i^T x_j$ is dot product

3. Decision rule (positive class: $w \cdot u + \beta_0 \geq 0$)

$$\left(\sum_{i=1}^{N} \alpha_i y_i x_i\right) \cdot u + \beta_0 \geq 0$$

u is the test data

decision is ≥ 0 instead of ≥ 1 as we care only about direction now

4. Solve for α_i in dual problem

Lagrange dual problem

$$\max_{\alpha} \frac{1}{2}\alpha^T \begin{pmatrix} y_1 y_1 x_1^T x_1 & \cdots & y_1 y_N x_1^T x_N \\ \vdots & \cdots & \vdots \\ y_N y_1 x_N^T x_1 & \cdots & y_N y_N x_N^T x_N \end{pmatrix} \alpha + (-1^T)\alpha$$

$$\Leftrightarrow \max_{\alpha} \sum_{i=1}^{N} \alpha - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \alpha \geq 0$$

Support Vector Classifier

Instead of using hard decision boundary, a soft margin is used

$$\max_{\beta, \beta_0, \epsilon} M$$

$$s.t. \sum_{j=1}^{p} \beta_j^2 = 1, y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ip}\right) \geq M(1 - \epsilon_i)$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C$$

1. C is a regularization parameter, larger C = more budget for error tolerance

Kernel formulation of SVM

Expanding to non-linear boundary

Expand linear boundary to non-linear boundary by replacing

$$x_i^T \cdot x_j \Rightarrow z_i^T \cdot z_j, z_i = \Phi(x_i)$$

$$\Rightarrow L_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j (z_i^T z_j)$$

1. Decision rule (positive class: $w \cdot z_u + \beta_0 \geq 0$)

$$\left(\sum_{i=1}^{N} \alpha_i y_i z_i\right) \cdot z_u + \beta_0 \geq 0$$

Kernel trick

Computing $z_i \cdot z_j$ directly is computational intensive, replace $z_i \cdot z_j$ with $K(x_i, x_j)$

$$x_i^T \cdot x_j \Rightarrow z_i^T \cdot z_j = K(x_i, x_j)$$

$$\Rightarrow L_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

Common kernels

polynomial of degree exactly d

$$K(x_i, x_j) = (x_i \cdot x_j)^d$$

polynomial of degree up to d

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d$$

Gaussian kernel (infinite features)

$$K(x_i, x_j) = exp\left(-\frac{\sum_{m=1}^{p}(x_{im} - x_{jm})^2}{2\sigma^2}\right)$$

$$= exp\left(-\frac{||x_i - x_j||_2^2}{2\sigma^2}\right)$$