Simple Linear Regression

1. $Y = \beta_0 + \beta_1 X + \varepsilon$

   β_0: intercept, β_1: slope

2. RSS: residual sum of squares

   $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$

   residual $e_i = y_i - \hat{y}_i$

   $RSS = \sum_i e_i^2$

   $= \sum_i (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$

   Goal: min RSS

3. Least square solution ☆

   $\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$

   $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$

4. null hypo H_0: $\beta_1 = 0$

   alternative hypo H_A: $\beta_1 \neq 0$

   t-statistic

   $t = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)}$ measures the no. of standard errors that $\hat{\beta}_1$ is away from 0.

   t-statistic follow t-distri with n-2 degrees of freedom.

   5% or 1%

   p-value is $2P(t_{n-2} \geq |t|)$. small pvalue → HA

5. Model accuracy assessment

   $R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$

   measures proportion of variability explained.

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

   $\log ab = \log a + \log b$

6. Transformation

   $Y \approx \gamma_0 e^{\gamma_1 X} \Rightarrow \log Y = \log \gamma_0 + \gamma_1 X$

Multiple Linear Regression

1. $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m$
2. Goal: min $\sum_i (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} \cdots - \hat{\beta}_m x_{im})^2$
3. 3 sorts of uncertainty with the prediction
   ① reducible errors: $\hat{\beta}_i$ are just estimations
   ② model bias: true model can be non-linear
   ③ irreducible errors: noise ε
4. dummy variable male/female → β_0 / $\beta_0 + \beta_1$

   polynomial regression $Y = \beta_0 + \beta_1 X + \beta_2 X^2$ still L model

Classification

predict qualitative response Y

1. most widely used classifier ① logistic regression ② KNN
   ③ Tree-based methods ④ Support vector machines

Logistic regression

$\hat{\beta}$, Odds, logit are positively correlated

1. $X = \{X_1, X_2 \cdots X_m\}$, $Y \in \{0, 1\}$

   $\hat{p}(X) = Pr(Y=1 \mid X)$

2. model: $\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_m X_m}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_m X_m}}$

3. Odds: $odds(X) = \frac{\hat{p}(X)}{1 - \hat{p}(X)} = e^{\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_m X_m}$

4. Log-odds (logit) = log(Odds) = $\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_m X_m$

5. given true values y_i, goodness of the fit

   choose $\hat{\beta}$ to maximize log-likelyhood

   $\max LL = \max \sum_{i:y_i=1} \log \hat{p}(x_i) + \sum_{i:y_i=0} \log(1 - \hat{p}(x_i))$

KNN: K nearest neighbours

1. for classification
   ① relabel training data set 从近到远
   ② prob that new point falls in class j using nearest K points

   $p_j(x) = \frac{1}{K} \sum_{i=1}^{k} I(y_i = j)$

   $I(y_i = j) = \begin{cases} 1 & \text{if } y_i = j \\ 0 & \text{otherwise} \end{cases}$ I: indicator function

   ③ predict $Y = k$, $k = \arg\max_{j=1\cdots k} p_j(x)$.

2. for regression: $y(x) = \frac{1}{k} \sum_{i=1}^{k} y_i$ ave KNN values

3. Error Rate & Accuracy Rate only for classification

   error rate $ER = \frac{\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)}{n}$

   note 3% AR flip it is better than 95% AR

   accuracy rate AR = 1 - ER → usually determine how good model is

Confusion matrix

Predicted level / Actual level

	P	N
P	TP	FP
N	FN	TN

1. Precision = $\frac{TP}{TP+FP}$, exactness
2. Recall = $\frac{TP}{TP+FN}$, completeness
3. F-score: $2 \times \frac{Precision \times Recall}{Precision + Recall}$, Large F-s → good model
4. Sensetivity = recall
5. Specficity = $\frac{TN}{TN+FP}$

MSE

$= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$ test MSE indicates test MSE - flexibility is a U-shape

Ridge Regression

$\hat{\beta}^R$

1. min $\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{m} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{m} \beta_j^2$
2. not scale equivalence $x_j \to c x_j \not\Rightarrow \beta_j^R = \frac{1}{c} \beta_j^R$

   usually $\beta_j \neq 0$ for all j unless $\lambda \to \infty$

Lasso Regression

$\hat{\beta}^L$

1. min $\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{m} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{m} |\beta_j|$
2. not scale equivalence, some β_j can be zero

Decision Tree

regression & classification tree

1. how to construct regions?

   min $RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$ mean of the region

2. Classification tree parameter (how good?) 3区小(越小越好)
   ① classification error rate $E = 1 - \max_k (\hat{p}_{mk})$
   ② Gini index $G = 1 - \sum_{k=1}^{K} \hat{p}_{mk}^2$
   ③ cross-entropy $D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$ 注:这里log base e

   Calculator ①. $\log_{10}(x)$ ② ÷ $\log_{10}(e)$.

Support Vector Machines

1. Maximal Margin Classifier

   $\max_{\beta_0 \cdots \beta_m} M$ subject to $\sum \beta_j^2 = 1$ ... ①

   $y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im}) \geq M \;\; \forall i = 1 \cdots n$ ... ② this is the distance from hyperplane

2. Support classifier: allow some violations

   $\max_{\beta_0 \cdots \beta_m, \varepsilon_1 - \varepsilon_n} M$ s.t. $\sum \beta_j^2 = 1$ ... ①

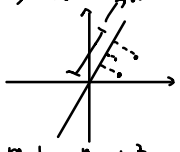
   $y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im}) \geq M(1 - \varepsilon_i) \; \forall i$ ... ②

   $\varepsilon_i \geq 0$, $\sum \varepsilon_i \leq C$ ... ③ violation budget

3. support vectors: on the margin, wrong side of margin or hyperplane
only S-V affects supporting hyperplane

Principle component Analysis

1. project multi-d display to 2-D PCA plot to cluster the data
2. $Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{m1}X_m$, $\sum \phi_j^2 = 1$
first PC has largest variance
$\phi_{11}, \cdots, \phi_{m1}$ → loadings



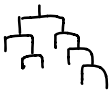
3. other PC are orthogonal to Z_1
4. proportion of variance explained: PVE
① total variance = $\sum_{j=1}^{m} Var(X_j) = \sum_{j=1}^{m} \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2$
② var explained = $\frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{m}\phi_j x_{ij})^2$
③ PVE = $\frac{\text{var explained}}{\text{total var}}$ ④ CPVE = $\sum_{I=1}^{M}$ PVE$_I$
⑤ $\sum_{i=1}^{\min(n-1,\,m)}$ PVE$_I$ = 1
5. how to find how many PC is suitable?
Look for elbow in CPVE plot. (以CPVE之后的PVE开始下降)

Clustering

K-Means Clustering

1. randomly 给每个点一个编号
2. iterate ①. 计算编号 K 的中心
②. assign 每个点 to 最近的中心
until no longer changes

Hirarchical Clustering



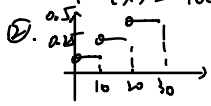
1. dendrogram. use similarity
2. ① 先找两两点之间的距离
② group 最近的2个, centroid 作为 class 的重心
③. iterate ①, ②.
3. complete linkage: 2个class中最远点之间 } sensitive to outlier
single: 2个class 最近
average: all pair-wise/n computational expensive
centroid: 重心之间 can cause inversion

Univariate optimization

1. $Z^* = \min F(X)$ subject to $a \le x \le b$; only consider $f(x)$ continuous in $[a,b]$
2. optimal solution: ① find points $f'(x)=0$ stationary point
② find $f(a)$ & $f(b)$ ③. find min among

Inventory models

1. Demand D, price p, cost c, savage s
$p > c > s$
2. profit = $\pi(y) = p\min(D,y) + s\max(y-D,0) - cy$
$= [(p-s)\min(D,y) + s\min(D,y)] + [s\max(y,D) - sD] - cy$
$= (p-s)\min(D,y) + s[\max(D,y) + \min(D,y)] - sD - cy$
$= (p-s)\min(D,y) + s(y+D) - sD - cy$
$= (p-s)\min(D,y) - (c-s)y$
$Z^* = \max \pi(y) = (p-s)E[\min(D,y)] - (c-s)y$
subject to $y \ge 0$
3. Solution $F(y^*) = \frac{p-c}{p-s}$ F is the CDF of D
if D continuous $y^* = F^{-1}(\frac{p-c}{p-s})$ → critical fractile 的 ↑ order more
overage cost $C_o = c-s$ buy more than demand
underage cost $C_u = p-c$ 少买少赚点钱

4. e.g. ① $F(x) = \frac{x}{100}$, $\alpha = 0.25$
$F^{-1}(x) = 100x$, $F^{-1}(0.25) = 25$
② 

10~20 可以

Linear Programming

1. $Z^* = \min(\max) \sum_{i=1}^{n} c_i x_i$
subject to $\sum_{i=1}^{n} a_{ji} x_i \le b_j$
2. to include $\ge$ constraint?
$\sum_i (-a_i) x_i \ge -b$
..."=" ? add both $-a \le -b$ & $a \le b$
3. sensitivity analysis
① small change ε in c_k ($c_k \to c_k + \varepsilon$
$Z^* \to Z^* + \varepsilon X_k^*$
② small change ε in b_j (RHS of constraint)
$Z^* \to Z^* + \varepsilon d_j$, d_j is the shadow price
1). 不closely bind 的 constraint 无 shadow price (dual val)
2). 在 constraint RHS 加1, 代入新 constraint 找 optimal val.

Linear integer programming

(在LP的基础上, allow some decision variable be int or binary