1 KNN

Training data $\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$, Estimate the $\hat{f}$

KNN: $\hat{y} = \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i$

Bias and Variance trade-off:

For a target point x_0 **Bias** of which: $f(x_0) - E_{D_n} \hat{f}(x_0)$

The variance of the estimator: $\mathbb{E}_{D_n}[(\hat{y} - \mathbb{E}_{D_n}[\hat{y}])^2] = \mathbb{E}[[\hat{f}(x_0) - \mathbb{E}\hat{f}(x_0)]^2]$

$$
\begin{aligned}
&\operatorname{Err}(x_0) \\
&= \mathbb{E}_{\mathcal{D}_n, Y_0}\left[\left(Y_0 - \widehat{f}(x_0)\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D}_n, Y_0}\left[\left(Y_0 - f(x_0) + f(x_0) - \mathbb{E}_{\mathcal{D}_n}\widehat{f}(x_0) + \mathbb{E}_{\mathcal{D}_n}\widehat{f}(x_0) - \widehat{f}(x_0)\right)^2\right] \\
&= \dots \\
&= \underbrace{\mathbb{E}_{Y_0}\left[(Y_0 - f(x_0))^2\right]}_{\text{Irreducible Error}} + \underbrace{\left(f(x_0) - \mathbb{E}_{\mathcal{D}_n}\widehat{f}(x_0)\right)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{\mathcal{D}_n}\left[\left(\widehat{f}(x_0) - \mathbb{E}_{\mathcal{D}_n}\widehat{f}(x_0)\right)^2\right]}_{\text{Variance}}
\end{aligned}
$$

Irreducible error cannot be reduced.

Model complexity: **1/k**

Degree of freedom: $df(\hat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(\hat{Y}_i, Y_i)$

1-NN df = n, k-NN df = n/k, n-NN df = 1

?? Why the variance of 1-nn is σ^2

Drawbacks:

- Need to store all training data
- Distance measure may affect the performance
- Curse of dimensionality: As the number of features increases, the volume of the space increases exponentially, making the data sparse. This sparsity makes it difficult to find neighbors that are close enough to be relevant.

2 Linear Regression

$$Y = X\beta + \epsilon$$

Loss function measrues the distance between the predicted values and the actual values.

$$Risk\ function\ R(f) := E(L(Y, f(X)))$$

$$\hat{f} = \arg\min_{f \in \mathcal{F}} R(f) = \arg\min_{f \in \mathcal{F}} E(L(Y, f(X))) = \frac{1}{n}\sum_{i=1}^{n} L(Y_i, f(X_i)) = \frac{1}{n}\sum_{i=1}^{n} (Y_i - f(X_i))^2$$

$$y_{n\times 1} = X_{n\times p}\beta_{p\times 1} + \epsilon_{n\times 1}$$

RSS: resuidual sum of squares $:= (y - X\beta)^T(y - X\beta)$
If X^TX is invertible, then the solution is given by:

$$\hat{\beta} = (X^TX)^{-1}X^Ty$$

$$Var(\hat{\beta}) = \sigma^2(X^TX)^{-1}$$

Residual: $r = \hat{e} = y - \hat{y} = (I - X(X^TX)^{-1}X^T)y$

$$\hat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n} r_i^2 = \frac{1}{n-p}(y - X\hat{\beta})^T(y - X\hat{\beta}) = \frac{1}{n-p}RSS$$

Among all unbiased linear estimators, the OLS estimator has the minimum variance.Further assuming ϵ is normal, $\hat{\beta}$ is also UMVUE.

1. $E(test\ error) = E(y^* - X\hat{\beta})^2 = E((y^* - X\beta + X\beta - X\hat{\beta})^2) = E(e^2) + Trace(X^TXCov(\hat{\beta})) = n\sigma^2 + p\sigma^2$
2. $E(training\ error) = E((I - H)y^2) = (n - p)\sigma^2$

Mallow C_p criterion:

$$C_p = \frac{RSS}{\sigma^2} + 2p - n$$

where p is the number of parameters in the model, n is the number of observations, and σ^2 is the variance of the errors.

3 Penalized Linear Regression

3.1 Ridge Regression

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \left((y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta\right)$$

An equivalent formulation is:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} (RSS)$$
$$subject\ to\ ||\beta||_2^2 \leq s$$

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

Bias of the ridge estimator $:= E(\hat{\beta}_{ridge}) - \beta = \dfrac{-\lambda}{1+\lambda}\beta_j =$ is biased.

$$Var = \frac{1}{(1+\lambda)^2} Var(\hat{\beta}_j^{OLS})$$

If we have orthogonal features, i.e., $X^T X = I$, then:

$$\hat{\beta}_{ridge} = \frac{1}{1+\lambda} X^T y$$

Understanding the shrinkage effect of ridge regression: P20 of the slides.

3.2 Lasso

$$\hat{\beta}_{lasso} = \arg\min_{\beta} \left((y - X\beta)^T (y - X\beta) + \lambda ||\beta||_1\right)$$

If we have orthogonal features, i.e., $X^T X = I$, then:

$$\hat{\beta}^{lasso} = \begin{cases} \hat{\beta}^{ols} - \lambda/2 & \text{if } ols > \lambda/2 \\ \hat{\beta}^{ols} + \lambda/2 & \text{if } ols < -\lambda/2 \\ 0 & \text{if } -\lambda/2 \leq ols \leq \lambda/2 \end{cases}$$

The equivalent formulation is similar to ridge regression with L1 norm.

Elastic Net:

$$\hat{\beta}_{elastic} = \arg\min_{\beta} \left((y - X\beta)^T (y - X\beta) + \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2^2\right)$$

4 Classification

$$y_i \in \{-1, 1\} or \{0, 1\}$$

$$0-1 Loss L(y, f(x)) := \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{if } o.w. \end{cases}$$

Soft classifier: estimate the conditional probability. Logistic
Hard classifier: estimate the class label directly. SVM

$$p(Y = y_i | X = x_i) = \eta(x_i)^{y_i}(1 - \eta(x_i))^{1-y_i}$$

$$where \log(\frac{\eta}{1-\eta}) = \beta^T x_i$$

log odds: $log\frac{p}{1-p}$, $\eta(x) = \frac{exp(x^\top \beta)}{1 + exp(x^\top \beta)}$

Use mle to estimate the parameters β:

$$l = \sum_{i=1}^{n} \log p(y_i | x_i, \beta)$$

Use newton's method to find the maximum likelihood estimate of β.

$$\beta^{new} = \beta^{old} - [\frac{\partial^2 l}{\partial \beta \partial \beta^\top}]^{-1} \frac{\partial l}{\partial \beta}|_{\beta^{old}}$$

Each unit increases in X_j increases the log-odds of Y by β_j

	Accept H_0	Reject H_0
H_0 true	✓	Type I Error
H_0 false	Type II Error	✓

Figure 1: Logistic Regression

第一类错误（假阳性）：健康人误诊为患者，相当于无罪的人被判有罪
第二类错误（假阴性）：患者漏诊为健康人，相当于有罪的人被判无罪

$$Overall\ Error = P(\text{假阳性}) + P(\text{假阴性}) = P(\text{假阳性}) + (1 - P(\text{真阳性}))$$

$$Sensitivity/Recall = P(\text{真阳性}) = \frac{TP}{TP + FN} = \frac{TP}{P(\text{患者})} = 1 - Type\ 2\ Error$$

$$Specificity = P(\text{真阴性}) = \frac{TN}{TN + FP} = \frac{TN}{P(\text{健康人})} = 1 - Type\ 1\ Error$$

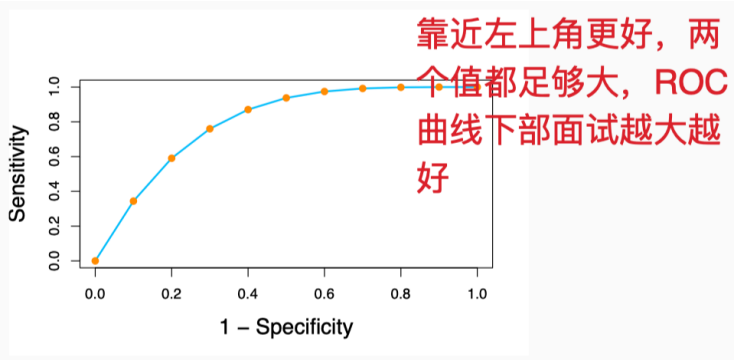

Figure 2: Chaneg different threshold to get different sensitivity and specificity

4.1 LDA and QDA

Bayes rule:

$$P(Y=1|X=x)=\frac{P(X=x|Y=1)P(Y=1)}{P(X=x)}$$

Treat $\pi=P(Y=1)$ as prior probabilities,and $f_1=P(X=x|Y=1)$ and $f_0=P(X=x|Y=0)$.

$$f_B(x)=arg\min_f R(f)=\begin{cases}1 & \text{if } f_1(x)\pi>f_0(x)(1-\pi)\\ 0 & \text{if } f_1(x)\pi<f_0(x)(1-\pi)\end{cases}$$

Multi-class

$$f_B(x)=arg\max_k P(Y=k|X=x)=arg\max_k \pi_k f_k(x)$$

Masking problem: linear polynomial may not perform well.

$$P(Y=k|X=x)=\frac{P(X=x|Y=k)P(Y=k)}{\sum_{j=1}^{K}P(X=x|Y=j)P(Y=j)}=\frac{f_k(x)\pi_k}{\sum_{j=1}^{K}f_j(x)\pi_j}$$

P26

$$\hat{f}(x)=arg\max_k log(\pi_k f_k(x))=arg\max -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)+log(\pi_k)$$

Note that we only care about terms related to k.
The discriminant function is:

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

then we could calculate the decision boundary by setting $\delta_k(x) = \delta_j(x)$ for $k \neq j$.

如何计算先验的概率 p31

QDA simply abandons the assumption of equal covariance matrices for all classes, allowing each class to have its own covariance matrix Σ_k.

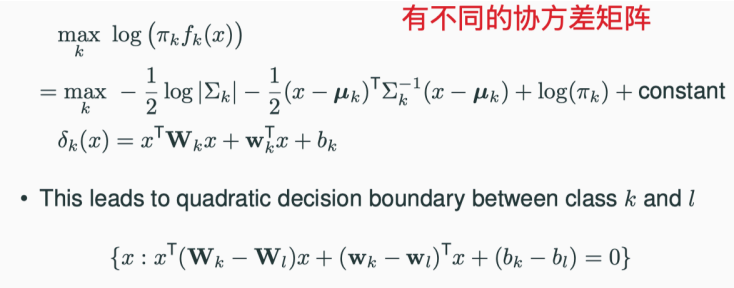

Figure 3: QDA decision boundary

Comparison of LDA and QDA:

- LDA assumes equal covariance matrices for all classes, while QDA allows each class to have its own covariance matrix.
- More paremeters in QDA
- Both are easy to implement.
- We could select quadratic terms and perform lda.
- p is large, the inverse of Σ might not exist.

4.2 Alternative Methods

Fisher criterion: maximize the ratio of between-class variance to within-class variance.

$$B = \sum_{k=1}^{K} \pi_k(\mu_k - \bar{\mu})(\mu_k - \bar{\mu})^\top$$

where $\bar{\mu} = \sum_{k=1}^{K} \pi_k \mu_k$ is the overall mean vector.

Denote the within class covariance matrix as W, which is common.

$$\max_a \frac{a^\top B a}{a \top W a}$$

The problem is equivalent to finding the eigenvector corresponding to the largest eigenvalue of the matrix $W^{-1}B$.

Regularization: sparse lda, **加一个** L1 penalty of vector a.
Regularized da (RDA): **在每个类的协方差矩阵上加一个** common **的矩阵。**
Naive Bayes: **把每个特征都作为独立的，** $f_k(x) \approx \Pi_{j=1}^{p} f_{kj}(x_j)$

Comparison between LDA and logistic regression:

- LDA assumes normality and equal covariance matrices, while logistic regression does not.
- LDA is a generative model, while logistic regression is a discriminative model.
- LDA can be more robust to outliers due to its assumptions.
- Logistic regression is easier to interpret in terms of odds ratios.

5 Support Vector Machines (SVM)

Support Vector Machines (SVM) are a powerful class of supervised learning algorithms used for classification and regression tasks. They work by finding the optimal hyperplane that separates different classes in the feature space.

$y_i \in \{-1, 1\}$

5.1 separable

Maximize the separation margin:

$$\begin{aligned} \max_{\boldsymbol{\beta}, \beta_0, \|\beta\|=1} \quad & M \\ \text{s.t.} \quad & y_i(x_i^\top \boldsymbol{\beta} + \beta_0) \geq M, \quad i = 1, \ldots, n \end{aligned}$$

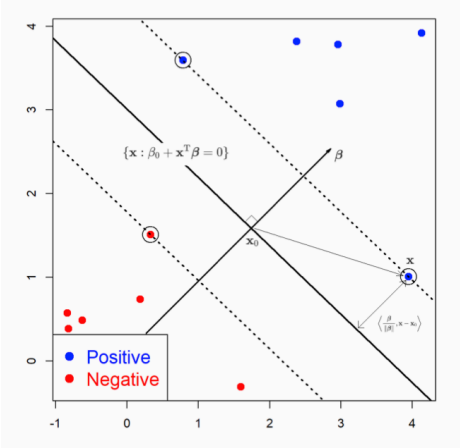

Figure 4: SVM with separable data

$$\min_{\boldsymbol{\beta}, \beta_0} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \text{ subject to } y_i(x_i^\top \boldsymbol{\beta} + \beta_0) \geq 1, \quad i = 1, \ldots, n$$

Consider the Lagrangian:

$$\mathcal{L}(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(x_i^\top \boldsymbol{\beta} + \beta_0) - 1)$$

Dual problem:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^\top x_j) \text{ subject to } \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, \ldots, n$$

The two problem are the same if 1. both g and h are convex, 2. the constraint h are feasible.

Advantages of the duality:

- SMO algorithms
- Change the problem from p dimension into n dimension.
- Kernel trick: we can use the kernel function to transform the data into a higher-dimensional space without explicitly computing the coordinates in that space.

- The SVM problem for separable case can be carried out as follows:
  - Solve dual for α_i's (those points for which $\alpha_i > 0$ are called "support vectors")
  - Obtain $\widehat{\boldsymbol{\beta}} = \sum_{i=1}^{n} \alpha_i y_i x_i$
  - Obtain β_0 by calculating the midpoint of two "closest" support vectors to the separating hyperplane

$$\widehat{\beta}_0 = -\frac{\max_{i:y_i=-1} x_i^{\mathsf{T}}\widehat{\boldsymbol{\beta}} + \min_{i:y_i=1} x_i^{\mathsf{T}}\widehat{\boldsymbol{\beta}}}{2}$$

  - For any new observation x, the prediction is

$$\text{sign}(x^{\mathsf{T}}\widehat{\boldsymbol{\beta}} + \widehat{\beta}_0)$$

线性可分的时候，logistic 的 likelihood 是正无穷，不好。
当满足高斯分布的时候，lda 是最好的，但是这个条件苛刻。

5.2 Nonseparable SVM

引入松弛变量

$$\begin{aligned} \min_{\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(x_i^\top\boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \quad i = 1,\ldots,n \\ & \xi_i \geq 0, \quad i = 1,\ldots,n \end{aligned}$$

C 越大，尽量分对。C 越小，尽量扩大间隔。

The Lagrangian is:

$$\mathcal{L}(\boldsymbol{\beta},\beta_0,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\nu}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(x_i^\top\boldsymbol{\beta} + \beta_0) - 1 + \xi_i) - \sum_{i=1}^{n}\gamma_i\xi_i$$

$\alpha_i, \gamma_i > 0$

Gradient：

$$\nabla_{\boldsymbol{\beta}}\mathcal{L} = \boldsymbol{\beta} - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\nabla_{\beta_0}\mathcal{L} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\nabla_{\boldsymbol{\xi}}\mathcal{L} = C - \boldsymbol{\alpha_i} - \boldsymbol{\gamma_i} = 0$$

Dual problem:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\gamma}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle \text{ subject to } \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, n$$

Useless data points are those with $\alpha_i = 0$.

Support vectors are those with $\alpha_i > 0$.

1. $0 < \alpha_i < C$ and $\xi_i = 0$
2. $\alpha_i = C$ and $\xi_i = 1 - y_i(x_i^\top \boldsymbol{\beta} + \beta_0) > 0$

5.3 Kernel Trick

我们只关心映射到另一个空间之后的样本之间的内积，而不是映射到另一个空间之后的样本本身。

The kernel trick allows us to compute the inner product in a high-dimensional space without explicitly mapping the data points to that space. This is particularly useful when dealing with non-linear decision boundaries.

The kernel function $K(x_i, x_j)$ is defined as:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

where $\phi(x)$ is a mapping function that transforms the input data into a higher-dimensional space.

Common kernel functions include:

- Linear kernel: $K(x_i, x_j) = x_i^\top x_j$
- Polynomial kernel: $K(x_i, x_j) = (x_i^\top x_j + c)^d$, where c is a constant and d is the degree of the polynomial.
- Radial basis function (RBF) kernel: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, where γ is a parameter that controls the width of the Gaussian function.

5.4 Convexity of SVM

$$\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) = \alpha^\top diag(y) K diag(y) \alpha$$

Convexity will be guaranteed if the Kernel matrix K is positive semi definite.

Mercer's theorem states that a kernel function is positive semi-definite if it can be expressed as an inner product in some feature space.

5.5 Penalized version of svm

Recall that the prime objective function for SVM is:

$$\min_{\boldsymbol{\beta},\beta_0} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t. } y_i(x_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \quad i = 1, \ldots, n \qquad \xi_i \geq 0, \quad i = 1, \ldots, n$$

$$\min_{\boldsymbol{\beta},\beta_0} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\max(0, 1 - y_i(x_i^\top \boldsymbol{\beta} + \beta_0))$$

Other forms of loss function:

1. Logistic loss: $log(1 + e^{-yf(x)})$
2. Modified Huber Loss: $L = \begin{cases} max(0, 1 - yf(x))^2, & \text{for } yf(x) \geq -1 \\ -4yf(x), & \text{o.w.} \end{cases}$

Some of which is differentiable, like logistic,modified huber, squared. Hinge loss is not differentiable at 0, but subgradient is available. 0,1 loss is not differentiable.

6 Tree and Random Forest

Nonparametric methods

6.1 Splitting rules

$$Gini(\mathcal{D}) = 1 - \sum_{k=1}^{K} p_k^2$$

$$Score = Gini(\mathcal{T}) - (\frac{N_{\mathcal{T}_L}}{N_{\mathcal{T}}} Gini(\mathcal{T}_L) + \frac{N_{\mathcal{T}_R}}{N_{\mathcal{T}}} Gini(\mathcal{T}_R))$$

Go through all variables j and all cutting points c to find the split with the best score.

$$(ID3, C4.5) Entropy = \sum_{k=1}^{N} -p_k log(p_k)$$

$$Error = 1 - max(p_k)$$

Gini and Shannon are more sensitive to changes. Maximum of $2^{M-1} - 1$ of possible splits.

For regression trees:

$$score = Var(\mathcal{T}) - (\frac{N_{\mathcal{T}_L}}{N_{\mathcal{T}}} Var(\mathcal{T}_L) + \frac{N_{\mathcal{T}_R}}{N_{\mathcal{T}}} Var(\mathcal{T}_R)) \qquad Var(\mathcal{T}) = \frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} (y_i - \bar{y})^2$$

For any sub-tree of T_{max} denote as $T \preceq T_{max}$ calculate

$$C_\alpha(T) = \sum_{\text{all termial nodes t in T}} N_t \cdot Impurity(t) + \alpha |T|$$

where $|T|$ is the number of terminal nodes in T, α is a penalty parameter.

$$\alpha \leq \frac{C(t) - C(T_t)}{|T_t| - 1}$$

逐渐删掉最小的 α

Tree methods can handle missing value by either putting them as a separate category or using surrogate variables.

Pros and cons of tree methods:

- Pros:
  - Easy to interpret and visualize.
  - Can handle both numerical and categorical data.
  - Non-parametric, no assumptions about the distribution of the data.
  - Can capture non-linear relationships.

- Cons:
  - Prone to overfitting, especially with deep trees.
  - Sensitive to small changes in the data.
  - Can be biased towards features with more levels (in categorical variables).
  - not smooth

6.2 Random Forest

不仅仅是多个树的叠加，feature **的选取也是随机的。**

Bootstrap

mtry: At each split, randomly select mtry variables from the entire set of features.

nmin: split until the number of samples in a node is less than nmin.

ntree: number of trees to grow.

Variable Importance For each tree m, use the oob as the testing set to obtain, obtain the prediction error: Err_0^m

For each variable j, permute the values of j in the oob set, and obtain the prediction error: Err_j^m.

$$VI_{mj} = \frac{Err_j^m}{Err_0^m} - 1$$

$$VI_j = \frac{1}{M}\sum_{m=1}^{M} VI_{mj}$$

- Formally,

$$\mathsf{VI}_{mj} = \frac{\sum_{i\in\mathcal{L}_m^o}\left[y_i - \widehat{f}(\mathbf{x}_i^{(-j)}, \tilde{x}_i^{(j)})\right]^2}{\sum_{i\in\mathcal{L}_m^o}\left[y_i - \widehat{f}(\mathbf{x}_i)\right]^2} - 1$$

  - $\mathcal{L}_m^o$ is the out-of-bag sample
  - $\widehat{f}$ is fitted using the in-bag sample $\mathcal{L}_m$
  - $\mathbf{x}_i^{(-j)}$ is the sub-vector of $\mathbf{x}$ by removing the jth entry $x_i^{(j)}$
  - $\tilde{x}_i^{(j)}$ is an independent copy from the marginal distribution of $x^{(j)}$
- Variable selection property
  - $\mathsf{E}[\mathsf{VI}_{mj}] \approx 0$ if $\widehat{f}$ is consistent and $x^{(j)}$ is irrelevant

Figure 5: Variable Importance

6.3 Adaptive Kernel

$$\hat{f}(x_0) = \frac{\sum_{i=1} y_i K_\lambda(x_0, x_i)}{\sum_{i=1} K_\lambda(x_0, x_i)}$$

In random forest, the kernel is adaptive to the local structure of the data. The bandwidth λ can be adjusted based on the density of points in the neighborhood of x_0.

$$\mathcal{A}(x_i, x_0) = \sum_{k \in \mathcal{K}} \mathbf{1}x_i \in A_k \cdot \mathbf{1}x_0 \in A_k = \begin{cases} 1 & \text{if } x_i \text{ and } x_0 \text{ are in the same terminal node} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Tree estimator } \hat{f}(x_0) = \frac{\sum_{i=1} y_i \mathcal{A}(x_i, x_0)}{\sum_{i=1} \mathcal{A}(x_i, x_0)}$$

$$\text{Random Forest estimator } \hat{f}(x_0) = \frac{\sum_{m=1}^{ntree} \sum_{i=1} y_i^m \mathcal{A}(x_i^m, x_0)}{\sum_{m=1}^{ntree} \sum_{i=1} \mathcal{A}(x_i^m, x_0)}$$

Random Forests is a kernel method with adaptive bandwidth:

- Each tree in a forest defines a kernel: uniform within each terminal node
- Ensemble: sum of kernels is still a kernel
- A tree is "more likely" to split on important variables, making their "bandwidth" smaller
- If the splitting variables are selected wisely, the bandwidth is adaptive to the signal strength
- The cutting point is more likely to happen on the zero curvature of the underlying target function

U statistics:

7 Boosting

7.1 Ada boost

$$\min_h \sum_{i=1}^{n} L(y_i, \sum_{k=1}^{t-1} f_k(x_i) + h(x_i))$$

Steps:

1. Initialize the weights: $w_i = \frac{1}{n}$ for all i.
2. For each iteration $t = 1, \ldots, T$:
3. Fit a weak learner h_t to the training data, minimizing the weighted loss:
4. $$h_t = \arg\min_h \sum_{i=1}^{n} w_i L(y_i, h(x_i))$$
5. Compute the error of the weak learner:
$$\epsilon_t = \sum_{i=1}^{n} w_i \mathbf{1}(y_i \neq h_t(x_i))$$
6. Compute the weight for the weak learner:
$$\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$
7. Update the weights for the next iteration:
$$w_i^{(t+1)} \leftarrow w_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))/Z_t$$
8. Final model is:
$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

7.2 Training Error Bound

$$\omega_i^{(T)} = \frac{1}{Z_{T-1}} \omega_i^{(T-1)} \exp(-\alpha_{T-1} y_i f_{T-1}(x_i))$$

$$\implies \omega_i^{(T+1)} = \frac{1}{Z_1 Z_2 \cdots Z_T} \omega_i^{(1)} \prod_{t=1}^{T} \exp(-\alpha_t y_i f_t(x_i)) = \frac{1}{Z_1 Z_2 \cdots Z_T} \frac{1}{n} exp[-y_i \sum_{t=1}^{T} \alpha_t f_t(x_i)]$$

Summation for all i:

$$1 = \sum_{i=1}^{n} \omega_i^{(T+1)} = \frac{1}{Z_1 Z_2 \cdots Z_T} \frac{1}{n} \sum_{i=1}^{n} exp[-y_i \sum_{t=1}^{T} \alpha_t f_t(x_i)]$$

$$\Longrightarrow Z_1 Z_2 \cdots Z_T = \frac{1}{n}\sum_{i=1}^{n} exp[-y_i \sum_{t=1}^{T} \alpha_t f_t(x_i)] = \frac{1}{n}\sum_{i=1}^{n} exp[-y_i F_T(x_i)] > \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{y_i \neq F_T(x_i)\}$$

By the definition of Z_t:

$$Z_t = exp[-\alpha_t] \sum_{y_i = f_t(x_i)} \omega_i^{(t)} + exp[\alpha_t] \sum_{y_i \neq f_t(x_i)} \omega_i^{(t)}$$

$$Z_t = (1-\epsilon_t) \cdot exp[-\alpha_t] + \epsilon_t \cdot exp[\alpha_t]$$

$$\frac{\partial Z_t}{\partial \alpha_t} = 0 \rightarrow \alpha_t = \frac{1}{2} log(\frac{1-\epsilon_t}{\epsilon_t})$$

$$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)} = \sqrt{1-4\gamma_t^2} \leq exp[-2\gamma_t^2]$$

Training error =

$$\sum_{i=1}^{n} \mathbf{1}\{y_i \neq F_T(x_i)\} < \sum_{i=1}^{n} exp[-y_i F_T(x_i)] = n Z_1 Z_2 \cdots Z_T \leq n \cdot exp[-2\sum_{t=1}^{T} \gamma_t^2] \rightarrow 0 \text{ for fixed n}$$

但是这不能说明 training error **是随** T **单调减的，只是他的上界是随** T **单调减的。**

Estimated probability:

$$E(exp[-yF(x)]) = e^{-F(x)} P(y=1|x) + e^{F(x)} P(y=-1|x)$$

Optimal $F^*(x)$ satisfies $\frac{\partial Loss}{\partial F} = 0$

$$F^*(x) = \frac{1}{2} log \frac{P(y=1|x)}{P(y=-1|x)} \qquad P(y=1|x) = \frac{exp\{2F^*(x)\}}{1+exp\{2F^*(x)\}}$$

8 Kernel

Histogram kernel:

$$\hat{f}(x) = \sum_{i=1}^{n} \frac{\mathbf{1}\{\mathbf{x_i} \in [\mathbf{x} - \lambda/\mathbf{2}, \mathbf{x} + \lambda/\mathbf{2}]\}}{\lambda \cdot n}$$

Bumpy and not smooth!

Denote K as a kernel function:

- $\int K(u)du = 1$
- $K(u) = K(-u)$
- $\int K(u)u^2 \leq \infty$
- $K_\lambda(u) = \frac{1}{\lambda}K(u/\lambda)$

Asymptotically unbiased for a target point x:

$$\begin{aligned}
\mathbb{E}[\widehat{f}(x)] &= \mathbb{E}\left[\frac{K\left(\frac{x-x_1}{\lambda}\right)}{\lambda}\right] \\
&= \int_{-\infty}^{\infty} \frac{1}{\lambda} K\left(\frac{x - x_1}{\lambda}\right) f(x_1)\, dx_1 \\
&= \int_{\infty}^{-\infty} \frac{1}{\lambda} K(t) f(x - t\lambda)\, d(x - t\lambda) \\
\text{(Taylor expansion)} &= f(x) + \frac{\lambda^2}{2} f''(x) \int_{-\infty}^{\infty} K(t)t^2\, dt + o(\lambda^2) \\
&\to f(x) \quad \text{as } \lambda \to 0
\end{aligned}$$

$$Integrated\ Bias^2 = \int (E[\hat{f}(x)] - f(x))^2 dx = \frac{\lambda^4 \sigma_K^4}{4} \int [f''(x)]^2 dx$$

$$where \quad \sigma_K^2 = \int K(t)t^2 dt$$

$$Integrated\ Var \approx \frac{1}{n\lambda} \int K^2(t) dt$$

8.1 Kernel Regression

K-nn **容易导致** boundary **上的样本有** bias due to asymmetry.

$$\hat{f}(x) = \frac{\sum\limits_i K_\lambda(x, x_i) y_i}{\sum\limits_i K_\lambda(x, x_i)}$$

Greater λ leads to smoother estimates, i.e., less variance but more bias.
Smaller λ leads to more variance but less bias.

$$\min_{\beta_0(x_0),\beta_1(x_0)} \sum_{i=1}^{n} (y_i - \beta_0(x_0) - \beta_1(x_0)x_i)^2 K_\lambda(x_0, x_i)$$

solution $\hat{f}(x_0) = \hat{\beta}_0(x_0) + \hat{\beta}_1(x_0) \cdot x_0$
这样权重自然就调整好了

Local Linear Regression Let $\boldsymbol{W}(x_0) = diag(K_\lambda(x_0, x_1), K_\lambda(x_0, x_2), \ldots, K_\lambda(x_0, x_n))$
Object: $l(\beta) = \boldsymbol{(y - X\beta)^\top W(x_0)(y - X\beta)}$
Solution is given by: $\hat{\beta} = (X^\top W X)^{-1} X^\top W y$

9 Clustering

9.1 CLuster Analysis

Group the dataset into subsets.

Flat clusteriing: assign into k cluster; Hierarchical clustering: arrange into a natural hierarchy.

Euclidian distance:

$$d(x,y) = \|u-v\|_2 = \sqrt{\sum_{i=1}^{p}(x_i-y_i)^2}$$

Hamming distance:

$$d(x,y) = \sum_{i=1}^{p} I(x_i \neq y_i)$$

Let $C(\dot{})$ be a cluster index function: $C : \{1,\cdot,n\} \to \{1,\cdot,K\}$

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i),C(i')=k} d(x_i,x_{i'})$$

Equivalent to maximizing the between cluster distance:

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i),C(i')\neq k} d(x_i,x_{i'})$$

9.2 Combinatorial Algorithm

Brute force search times(prove by induction):

$$S(n,K) = \frac{1}{K!}\sum_{i=1}^{K}(-1)^{K-i}\binom{K}{i}i^n$$

9.3 K-mean

$$\min_{C,\{m_k\}_{k=1}^{K}} \sum_{k=1}^{K}\sum_{C(i)=k}\|x_i-m_k\|^2$$

NP-hard for ≥ 2 dimensions.

Steps:

1. Fixing C, find the best m_k: $m_k = \dfrac{\sum_{C(i)=k} x_i}{\sum_i \mathbf{1}\{C(i)=k\}}$

2. Fixing m_k, find the best C: $C(j) = argmin_i d(x_j, m_i)$

3. Repeat 1 and 2 until convergence.

K-medoids: Replace the second step with searching for the **1 observation** that minimizes the within cluster distance, better for categorical mission.

9.4 Hierarchical Clustering

1. Begin with n clusters, each containing one observation.

2. Find the two clusters that are closest together, and merge them into a single cluster.

3. Repeat step 2 until there is only one cluster left.

Distance between two clusters:

- Single linkage: $d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$
- Complete linkage: $d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$
- Average linkage: $d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$
- Centroid linkage: $d(C_i, C_j) = d(\bar{x}_i, \bar{x}_j)$ where $\bar{x}_k$ is the centroid of cluster C_k.

Dissimilarity matrix: 1.symmetric, 2.diagonal is 0

9.5 Spectral Clustering

Adjacency matrix W: w_{ij} is the similarity between x_i and x_j.

Degree matrix D as a diagonal matrix: $d_{ii} = \sum_{j=1}^{n} w_{ij}$.

Laplacian matrix $L = D - W$.

Normalized Laplacian matrix:

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

The eigenvalues of L_{sym} are non-negative, and the smallest eigenvalue is 0.

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

- Compute the smallest k eigenvalue of L, denote them collectively as $V_{n \times k}$
- Treat $V_{n \times k}$as the matrix of the observed data, and perform k-means clustering
- Output the k cluster labels

10 Survival Analysis

10.1 preliminary

1. Survival function: $S(t) = P(T > t)$, where T is the time until the event of interest occurs.
2. Hazard function: $h(t) = \lim_{\Delta t \to 0} \frac{P(t<T\leq t+\Delta t|T\geq t)}{\Delta t}$, which describes the instantaneous risk of the event occurring at time t given that it has not occurred before t.
3. Cumulative hazard function: $H(t) = \int_0^t h(u)du$, which accumulates the hazard over time.
4. $f(t) = h(t)S(t) = -S'(t)$ is the probability density function of the time until the event occurs.

$$H(t) = \int_0^t h(u)du = \int_0^t \frac{f(u)}{S(u)}du = \int_0^t \frac{1}{S(u)}d(-S(u)) = -\ln S(t)$$
$$H(t) \sim EXP(1)$$

$$T = min(T_1, T_2, \ldots, T_n) \to h^T(t) = \sum_{i=1}^{n} h_i(t)$$

Expotential distribution: $S(t) = e^{-\lambda t}$, where λ is the rate parameter. $f(t) = \lambda e^{-\lambda t}, h(t) = \lambda, H(t) = \lambda t$

Weibull distribution: $S(t) = e^{-(\lambda t)^p}$, where λ is the scale parameter and k is the shape parameter. $f(t) = \lambda p t^{p-1} e^{-(\lambda t)^p}, h(t) = \lambda p t^{p-1}, H(t) = (\lambda t)^p$

Homogeneous Poisson process: $h(t) = \lambda$, where λ is the rate of the process.

$$N(t+s) - N(t) \sim Poisson(\lambda s)$$

Censoring Censoring occurs when the event of interest has not occurred by the end of the observation period. There are two types of censoring:

1. Type I censoring:

$$(U_i, \delta_i) = \{min(T_i, c), 1(T_i \leq c)\}$$

2. Type II censoring: Only observe the first r smallest survival times.

$$T_{(1,n)}, T_{(2,n)}, \ldots, T_{(r,n)}$$

3. Random censoring:

$$(U_i, \delta_i) = \{min(T_i, C_i), 1(T_i \leq C_i)\}, C_i \sim iid$$

4. Interval censoring: observe only (L_i, U_i)

Noninformative censoring:

$$h(t) = \lim_{\epsilon \to 0} \frac{P(t \le T \le t + \epsilon | T \ge t, C \ge t)}{\epsilon}$$

Likelihood construction:

$$L_i(F, G) = \begin{cases} f(u_i)(1 - G(u_i)), & \text{if } \delta_i = 1 \\ S(u_i)g(u_i), & \text{if } \delta_i = 0 \end{cases}$$

$$L(F, G) = \prod_{i=1}^{n} L_i(F, G) = \prod_{i=1}^{n} [f(u_i)^{\delta_i} S(u_i)^{1-\delta_i}][g(u_i)^{1-\delta_i} G(u_i)^{\delta_i}]$$

$$L(F) = \prod_{i=1}^{n} [f(u_i)^{\delta_i} S(u_i)^{1-\delta_i}] = \prod_{i=1}^{n} [h(u_i)^{\delta_i} S(u_i)]$$

10.2 Kaplan-Meier estimator

If no censoring,

$$\hat{S}(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(T_i > t)$$

$T_1 < T_2 < \cdots < T_n$, $f_1 = P(T = T_1), f_2 = P(T = T_2)$

- Obs: $2, 2, 3^+, 5, 5^+, 7, 9, 16, 16, 18^+$, where $^+$means censored
- $v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 7, v_5 = 9, v_6 = 16, v_7 = 18, v_8 = 18^+$
- The likelihood function in terms of $(f_1, f_2, \cdots)$:
  $L(F) =$
  $f_1^2 (f_3 + f_4 + f_5 + f_6 + f_7 + f_8) f_3 (f_4 + f_5 + f_6 + f_7 + f_8) f_4 f_5 f_6^2 f_8$,
  where $f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 = 1$

Figure 6: example,p31

Too complicated!!!
Reparametrization tricks:

$$h_1 = P(T = v_1), h_j = P(T = v_j | T > v_{j-1}), j = 2, \ldots, n$$

- For $t \in [v_j, v_{j+1})$

$$S(t) = P(T > t) = P(T > v_j) = \prod_{i=1}^{j} (1 - h_i)$$

- For $t = v_j$

$$f_j = f(t) = P(T = t) = h_j \prod_{i=1}^{j-1}(1 - h_i)$$

The likelihood function in terms of $(h_1, h_2, \cdots)$:

$$\begin{aligned} L(F) =& h_1^2 \times \{(1-h_1)(1-h_2)\} \times \{(1-h_1)(1-h_2)h_3\} \\ & \times \{(1-h_1)(1-h_2)(1-h_3)\} \times \{(1-h_1)(1-h_2)(1-h_3)h_4\} \\ & \times \{(1-h_1)(1-h_2)(1-h_3)(1-h_4)h_5\} \\ & \times \{(1-h_1)(1-h_2)(1-h_3)(1-h_4)(1-h_5)h_6\}^2 \\ & \times \{(1-h_1)(1-h_2)(1-h_3)(1-h_4)(1-h_5)(1-h_6)(1-h_7)\} \\ =& h_1^2(1-h_1)^8 \times (1-h_2)^8 \times h_3(1-h_3)^6 \\ & h_4(1-h_4)^4 \times h_5(1-h_5)^3 \times h_6^2(1-h_6) \times (1-h_7) \end{aligned}$$

Figure 7: solution to Figure 6, p32

$$L(F) = \prod_{j=1}^{n} [h_j^{d_j}(1-h_j)^{Y(v_j)-d_j}]$$

where d_j is the number of events at time v_j, and $Y(v_j)$ is the number of individuals at risk at time v_j.

d_j： v_j 时间嗝屁的人； $Y(v_j) - d_j$ 这个时间幸存的人

Thus, we could derive the Kaplan-Meier estimator as follows:

$$\hat{h}_j = \frac{d_j}{Y(v_j)}, \quad j = 1, \ldots, n$$

$$\hat{S}(t) = \begin{cases} 1 & t < v_1 \\ \prod_{j=1}^{k}(1-\hat{h}_j) & v_k \le t < v_{k+1} \end{cases}$$

1. If the last $d_g = Y(v_g)$, then $\hat{S}(t) = 0$ for $t \ge v_g$
2. If the last $d_g < Y(v_g)$, then $\hat{S}(t) > 0$ but not defined for $t > v_g$.

Calculate the example above

Nelson-Aalen Estimator　在没有解析解的时候很有用

- No censoring $\hat{S}(t) = n^{-1} \sum_{i=1}^{n} I(T_i > t)$
- Right censoring: $\hat{S}(t) = n^{-1} \sum_{i=1}^{n} E(I(T_i > t) \mid U_i, \delta_i)$
  1. $E(I(T_i > t) \mid U_i, \delta_i = 1) = I(U_i > t)$
  2. $E(I(T_i > t) \mid U_i, \delta_i = 0) = S(t)/S(U_i) I(t \geq U_i) + I(U_i > t)$
- Self-consistency iteration:

$$\hat{S}_{new}(t) = n^{-1} \sum_{i=1}^{n} \left\{ I(U_i > t) + (1 - \delta_i) \frac{\hat{S}_{old}(t)}{\hat{S}_{old}(U_i)} I(U_i \leq t) \right\}$$

- The solution is still the KM estimator.

Figure 8:

$$\hat{H}(t) = \sum_{j=1}^{n} \frac{d_j}{Y(v_j)} \mathbf{1}(v_j \leq t) = \sum_{i=1}^{j} \hat{h}_i \mathbf{1}(v_i \leq t)$$

$$H(t) = -\ln \hat{S}(t) = \sum_{i=1}^{j} -\ln(1 - \hat{h}_i) \mathbf{1}(v_i \leq t) \approx \sum_{i=1}^{j} \hat{h}_i \mathbf{1}(v_i \leq t)$$

$$n \to \infty, \hat{S}(t) \xrightarrow{p} S(t) \qquad \sqrt{n}\{\hat{S}(t) - S(t)\} \xrightarrow{d} N(0, \sigma^2(t))$$

$$Var(\hat{h}_i) = \frac{\hat{h}_i}{(1 - \hat{h}_i) Y(v_i)}$$

方差估计考试不涉及

AUC $\mu = \int_0^\tau S(t)dt = tS(t)|_0^\tau + \int_0^\tau tf(t)dt = \tau S(\tau) + \int_0^\tau tf(t)dt$
$= \int_0^\infty min(t, \tau) f(t)dt = E\{min(T, tau)\}$

Restricted mean survival time

- The restricted mean survival time $E\{\min(T,\tau)\}$ can also be estimated as

$$\hat{\mu}_{IPW} = n^{-1}\sum_{i=1}^{n}\frac{\delta_i + (1-\delta_i)\, I\,(U_i \geq \tau)}{\widehat{S}_C\,(T_i \wedge \tau)}\, T_i \wedge \tau$$

  where $\widehat{S}_C(\cdot)$ is a consistent estimator of the survival function of the censoring time C.

- Rational

$$E\left[\frac{I\,(C_i \geq \tau \wedge T_i)}{\widehat{S}_C\,(T_i \wedge \tau)}\, T_i \wedge \tau \mid T_i\right] \approx (T_i \wedge \tau)\,\frac{P\,(C_i \geq \tau \wedge T_i \mid T_i)}{S_C\,(T_i \wedge \tau)} = T_i \wedge \tau$$

- This type of estimator is called the inverse probability weighting estimator

Figure 9: estimating the censoring time, p50

10.3 Logrank Test

$\tau_1 < \tau_2 < \cdots < \tau_K$

$Y_i(\tau_j)$: the number of individuals at risk at time τ_j in group i.

d_{ij}: the number of events at time τ_j in group i.

超几何分布:

$$P(d_{1j} = d) = \frac{\binom{d_j}{d}\binom{Y(\tau_j)-d_j}{Y_1(\tau_j)-d}}{\binom{Y(\tau_j)}{Y_1(\tau_j)}}$$

$$E(d_{1j}) = \frac{d_j Y_1(\tau_j)}{Y(\tau_j)} \qquad Var(d_{1j}) = \frac{d_j Y_1(\tau_j) Y_0(\tau_j)(Y(\tau_j) - d_j)}{Y(\tau_j)^2 (Y(\tau_j) - 1)}$$

$O_j = d_{1j}$ observed number of events in group 1 at time τ_j

$E_j = \frac{d_j Y_1(\tau_j)}{Y(\tau_j)}$ expected number of events in group 1 at time τ_j

$V_j = \frac{d_j Y_1(\tau_j) Y_0(\tau_j)(Y(\tau_j) - d_j)}{Y(\tau_j)^2 (Y(\tau_j) - 1)}$ variance of O_j

Logrank test statistic:

$$Z = \frac{\sum_{j=1}^{K}(O_j - E_j)}{\sum_{j=1}^{K}\sqrt{V_j}} \sim N(0,1)$$

The power of the log rank test depends on the number of observed failures rather than the sample sizes

$$\sum_{j=1}^{K}(O_j-E_j)=\sum_{j=1}^{K}\frac{Y_0(\tau_j)T_1(\tau_j)}{Y(\tau_j)}(\frac{d_{1j}}{Y_1(\tau_j)}-\frac{d_{0j}}{Y_0(\tau_j)})=\sum_{j=1}^{K}\frac{Y_0(\tau_j)T_1(\tau_j)}{Y(\tau_j)}(\hat{h}_{1j}-\hat{h}_{0j})=\int_0^{\infty}\frac{Y_0(t)T_1(t)}{Y(t)}d\{\hat{H}_1(s)-\hat{H}_0($$

10.4 Cox

Cox model: $h(t|Z=z)=h_0(t)g(z)$

$$\frac{h(t|Z=z_1)}{h(t|Z=z_2)}=\frac{h_0(t)g(z_1)}{h_0(t)g(z_2)}=\frac{g(z_1)}{g(z_2)}$$

Observed data: $(U_i,\delta_i,Z_i)=\{min(T_i,C_i),1(T_i\le C_i)\}$
Noninformative censoring: $T_i,C_i|Z_i$

$$L(\beta,h_0)=\prod_{i=1}^{n}L_i(\beta,h_0)=\prod_{i=1}^{n}[h_0(u_i)^{\delta_i}g(Z_i)^{\delta_i}S_0(u_i)]$$

where $S_0(t)=e^{-\int_0^{u_i}h_0(t)g(z)dt}$ is the baseline survival function.
$\mathcal{F}(t_j^-)$: Information at time t_j before the event occurs.
$\mathcal{I}(t)$: Set of subjects failed at time t

$$\begin{aligned}
L(\beta,h_0(\cdot))&=PL(\beta)\tilde{L}(\beta,h_0(\cdot))\quad\text{where}\\
PL(\beta)&=\text{pr}\left(\mathcal{I}(\tau_1)\mid\mathcal{F}(\tau_1^-),h_0,\beta\right)\times\text{pr}\left(\mathcal{I}(\tau_2)\mid\mathcal{F}(\tau_2^-),h_0,\beta\right)\\
&\quad\cdots\text{pr}\left(\mathcal{I}(\tau_K)\mid\mathcal{F}(\tau_K^-),h_0,\beta\right)\\
&=\text{pr}\left(\mathcal{I}(\tau_1)\mid\mathcal{F}(\tau_1^-),\beta\right)\times\text{pr}\left(\mathcal{I}(\tau_2)\mid\mathcal{F}(\tau_2^-),\beta\right)\\
&\quad\cdots\text{pr}\left(\mathcal{I}(\tau_K)\mid\mathcal{F}(\tau_K^-),\beta\right)\\
\text{where}\quad&\text{pr}\left(\mathcal{I}(\tau_j)\mid\mathcal{F}(\tau_j^-),h_0,\beta\right)\\
&=\text{pr}\left(\text{subject }j_1\text{ fails at }\tau_j\mid\text{subjects }j_1,\cdots,j_p\text{ survive at }\tau_j^-\text{and one fails at }\tau_j\right)
\end{aligned}$$

$$\text{pr}\left(\text{subject }j_1\text{ fails at }\tau_j\mid\text{subjects }j_1,\cdots,j_p\text{ survive at }\tau_j^-\text{and one fails at }\tau_j\right)$$

$$=\frac{g(z_{j_1})}{\sum_{i=1}^{p}g(z_{j_i})}$$

Go back to the example

$$PL(\beta)=\prod_{i=1}^{n}(\frac{e^{\beta' z_i}}{\sum_{j=1}^{n}1(u_j\ge u_i)e^{\beta' z_j}})^{\delta_i}$$

Profile likelihood：**先对** h **求导，得出一个含有** beta **的** h **的解。先对** L **做一个** log，**然后求导算出** h_i，**带回到** L **里面，发现和** partial likelihood **正比。**
Stratified Cox model：relax **了部分条件，允许不同组别有不同的** h_k

11 Causal Inference

Defining causal quantities, this will be done in terms of counterfactuals.

Stating assumptions necessary to identify causal quantitiees.

Defining a mathematical model to deal with the curse of dimensionality.

Aspirin has no effect if $Y^1 = Y^0$

$$Y = Y^1 \cdot A + Y^0 \cdot (1 - A)$$

$$ATE : \psi = E[Y^1 - Y^0] = E[Y^1|A = 1] - E[Y^0|A = 0] = E[Y|A = 1] - E[Y|A = 0]$$

- CA consistency assumption $Y = Y^A$ w.p. 1
- RA randomization assumption $A \perp (Y^1, Y^0)$
- PA positivity assumption $0 < P(A = 1) < 1$

$E(Y^a) = E(T|A = a) = \sum_l E(Y|A = a, L = l) f_L(l|a)$
$= \sum_l E(Y|A = a, L = l) f_L(l)$ if $L \perp A$

NUCA no unmeasured confounding assumption: $(Y^1, Y^0) \perp A|L$

positivity $f_L(l) > 0 \rightarrow P(A = a|L = l) > 0$

$$E(Y^a) = \sum_l E(Y|A = a, L = l) P(L = l) \text{ or } \int E(Y|A = a, L = l) f_L(l) dl$$

$$\begin{aligned} E(Y^a) &= E(E(Y^a|L)) \\ &= \sum_l E(Y^a|L = l) f_L(l) \\ &= \sum_l E(Y^a|A = a, L = l) f_L(l) \quad \text{(NUCA)} \\ &= \sum_l E(Y|A = a, L = l) f_L(l) \quad \text{(CA)} \end{aligned}$$

G-formula

Assume both A and L are categorical variables with low to moderate number of levels, so that $b(a, l)$ is given by the stratified sample average:

$$\hat{b}(a, l) = \frac{\sum_{i=1}^n I(A_i = a, L_i = l) Y_i}{\sum_{i=1}^n I(A_i = a, L_i = l)}$$

and

$$\hat{f}_L(l) = \frac{1}{n} \sum_{i=1}^n I(L_i = l)$$

The nonparametric estimator of the G-formula is given by:

$$\begin{aligned}\hat{g}(a) &= \sum_l \hat{b}(a,l)\hat{f}_L(l) \\ &= \sum_l \hat{b}(a,l)\frac{1}{n}\sum_{i=1}^{n} I(L_i = l) \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_l \hat{b}(a,l)I(L_i = l) \\ &= \frac{1}{n}\sum_{i=1}^{n} \hat{b}(a,L_i)\end{aligned}$$

Crude association estimator:

$$E(Y|A=a)$$

$$IPTW = E[Y^a] = E[\frac{1(A=a)}{f(A|L)}Y]$$

12 Appendix

v_j	$Y(v_j)$	d_j	$\hat{h}_j$	$\hat{S}(v_j) = \prod_{i=1}^{j}(1-\hat{h}_i) = \hat{P}(T > v_j)$
2	10	2	2/10	.8
5	7	1	1/7	.69 $(= .8 \times \frac{6}{7})$
7	5	1	1/5	.55 $(= .69 \times \frac{4}{5})$
9	4	1	1/4	.41 $(= .55 \times \frac{3}{4})$
16	3	2	2/3	.14 $(= .41 \times \frac{1}{3})$
18	1	0	0	.14

Figure 10: example for km, p36

Go back to Kaplan-Meier

- Data: $(U_i, \delta_i, Z_i) = (21,1,1), (16,0,0), (13,0,1), (12,1,0), (11,1,1)$
- $\tau_1, \tau_2, \tau_3 = 11, 12, 21$
- $R_1 = \{1,2,3,4,5\}, \quad R_2 = \{1,2,3,4\}, \quad R_3 = \{1\}$
- The partial likelihood has three terms:

$$PL(\beta) = \left(\frac{e^{\beta}}{3e^{\beta}+2}\right)\left(\frac{1}{2e^{\beta}+2}\right)\left(\frac{e^{\beta}}{e^{\beta}}\right)$$

Figure 11:

Go back to Cox