CM763 - Classification Summary

Definition 0.1 (Bias)**.** Is difference between expected value of dist. of $\hat{\theta}$ and true, $\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta$. Low bias: Trees, k-NN, SVM; High bias : LinR, LogR

Definition 0.2 (Prior Distribution)**.** $\pi_k = Pr(Y = k)$

Definition 0.3 (Cond. Distrib. of x given y)**.** $f_{x|y}(X|Y = k) = f_k(x)$ (probability density)

Definition 0.4 (Bayes' Rule)**.** $Pr(Y = k|X = x) = \frac{Pr(X=x|Y=k)Pr(Y=k)}{Pr(X=x)} = \frac{f_k(x)\pi_k}{\sum_{j=1}^{k} \pi_j f_j(x)}$; $P(A|B) = \dfrac{P(A)P(B|A)}{P(B)}$

Definition 0.5 (Bayes Classifier)**.** $f(x) := \text{argmax}_k Pr(Y = K|X = x)$, or $C(x) = j$ if $P_j(x) = \max\{P_1(x), \ldots, P_K(x)\}$. Makes fewest mistakes

Definition 0.6 (Joint Distribution)**.** $Pr(X, Y) = Pr(X|Y)Pr(Y)$, $X \in \mathbb{R}^d$

Definition 0.7 (Naive Bayes)**.** Assumes $Pr(X|Y) = Pr(X_1|Y) \cdot \Pr(X_2|Y) \ldots Pr(X_d|Y)$

Definition 0.8 (Log-odds)**.** $\log\left(\frac{p}{1-p}\right)$

Definition 0.9 (Parametric)**.** Can be determined up to finite number of parameters. If finite number of parameters are known, the posterier distribution can be known. Structure is fixed

Definition 0.10 (Nonparametric)**.** Non-deterministic with a finite number of parameters. If finite number of parameters are known, the posterier distribution still cannot be known. Can grow without bound as data increases

Definition 0.11 (Generative Classifier)**.** Models the distribution of input characteristics of the class (e.g. Naive Bayes Classifier). A Generative Model learns the joint probability distribution $Pr(x, y)$; it predicts the conditional probability with the help of Bayes Theorem.

Definition 0.12 (Discriminant Classifier)**.** Tries to estimate parameters of decision boundary/class separator directly from labelled data. Models $P_k(x) = P(Y = k|X = x)$ directly (e.g. LogR).

Definition 0.13 (Geometric Mean)**.** For n numbers, multiply them all together and then take the n^{th} root: $\sqrt[n]{a_1 a_2 \ldots a_n}$

Trade-offs: (1) Prediction Accuracy vs Interpretability; (2) Good fit vs Underfit/Overfit (3) Parsimony vs Complex. **Sensitivity**: commonly used to validate the accuracy of a classifier; Predicted True/Total events. **Selection Bias**: when sample obtained not representative of population

1 Linear Classifiers

1.1 Linear Regression

[*parametric*] A linear relationship between response and explanatory variables, i.e. $y = \beta_0 + \beta_1 x + \epsilon$. If you want to estimate probabilities, use LR. If > 2 categories, LinR not appropriate since arbitrary numeric values to categories $\implies$ bigger "difference" $\implies$ use Discrim. Analysis

1.2 Gaussian Classifiers: LDA & QDA

Assumes Gaussian distribution for densities, $X|Y = k \sim N_p(\mu_k, \Sigma_k)$. $f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right)$. Discr. Analysis models the distribution of X in each class separately, then uses Bayes Thm to obtain $Pr(Y|X)$

Definition 1.1 (Discriminant Variables)**.** linear combinations of features

1.2.1 Linear Discriminant Analysis

LDA assumes Σ_k all equal. Compute discriminant function for each class then classify to largest. $G(x) = \text{argmin}_k \delta_k(x)$ where $\delta_k(x) = \log f_k(x)\pi_k = \log \pi_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + x^T\Sigma^{-1}\mu_k$ (discriminant function). Each discriminant is a linear combination of predictors. π_k & μ_k can be est. from sample. Σ can be estimated by "pooled variance": $\hat{\Sigma} = \frac{(x-\hat{\mu})^T(x-\hat{\mu})}{n-K}$

PCA vs LDA: PCA (unsupervised) finds axis of maximal variance. LDA finds a feature-space that maximizes class separability

1.2.2 Quadratic Discriminant Analysis

QDA does not assume Σ_k are equal.

1.3 Logistic Regression

[*discriminative/parametric*] $p(x) = \dfrac{e^{\beta_0+\beta\cdot x}}{1+e^{\beta_0+\beta\cdot x}}$. Equiv. to Logit: $\frac{P_i(x)}{1-P_i(x)} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ (to model categorical variables). Predict binary outcome

Multinomial: $Pr(Y = k|X) = \dfrac{e^{\beta_{0k}+\beta_{1k}X_1+\cdots+\beta_{pk}X_p}}{\sum_{l=1}^{K} e^{\beta_{0l}+\beta_{1l}X_1+\cdots+\beta_{pl}X_p}}$

Logistic with L_1-Regularization: $Q(\beta_0, \ldots, \beta_p) = l(\beta_0, \ldots, \beta_p) + \lambda\sum_{j=1}^{p}|\beta_j|$, w/ $\lambda = \infty \implies$ all β_j's zero. As $\lambda \downarrow$ from ∞, most important β_j nonzero first, then second, etc.

1.4 LASSO: L_1 Regularization

$\sum_{i=1}^{n}(y_i - \beta_0 - \sum x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$. Min Opt problem essentially objective w/ $\sum_{j=1}^{p}|\beta_j| \leq t$ constraint. See Section 1.3 for zeroing β_j's

1.5 Ridge Regression: L_2 Regularization

Reduces variance by adding Bias. Performs better when most variables useful. Tuning parameter λ. Penalizes slope

1.6 AIC & BIC

Smaller the better. BIC penalizes complex models more than AIC. BIC asymptotically consistent in model selection. AIC tends to choose more complex as $n \to \infty$

2 Nonparametric Classifiers

2.1 K-Nearest Neighbors

[*discriminative*] large K: low variance, high bias; smaller K: high variance, low bias. Struggles in high dimensions. Classifies using majority vote among K neighbors

2.2 Smooth Binomial Regression

[*nonparmetric*] $\log\left(\frac{P_1(x)}{1-P_1(x)}\right) = f(x_1, \ldots, x_p)$, a smooth function (e.g. use splines). But suffers curse of dim.

2.3 Kernel Density Classification

$\hat{f}(x_0) = \frac{1}{n\lambda}\sum_{i=1}^{n} K_\lambda(x_0, x_i) = \frac{1}{n\lambda}\sum_{i=1}^{n} K\left(\frac{x_0-x_i}{\lambda}\right)$

2.4 Naive Bayes Classifier

[*Indep. Feature Model*]. Apply a flexible model of the ratio of conditional densities directly. In discriminant analysis:

$$\log\left[\frac{P(Y = k|\mathbf{X} = \mathbf{x})}{P(Y = K|\mathbf{X} = \mathbf{x})}\right] = \log\frac{\pi_k}{\pi_K} + \log\frac{f_k(\mathbf{x})}{f_K(\mathbf{x})}$$

Naive Bayes *assumes independence of the X_j's.* (e.g. assumes $Pr(X|Y) = Pr(X_1|Y) \cdot \Pr(X_2|Y) \ldots Pr(X_d|Y)$). Thus our ratio becomes $\log\dfrac{\pi_k}{\pi_K} + \log\dfrac{f_k(x)}{f_K(x)} = \log\dfrac{\pi_k}{\pi_K} + \displaystyle\sum_{j=1}^{p}\log\frac{f_{kj}(x_j)}{f_{Kj}(x_j)}$

$= \alpha_k + \displaystyle\sum_{j=1}^{p} h_{kj}(x_j)$ Each $h_{kj}(x_j)$ is a log-ratio of densities for X_j between class k and class K. PCA retains the full information of the space X, but re-expresses it as uncorrelated components. Running PCA before naive Bayes can sometimes *substantially* improve performance.

2.5 Generalized Additive Model

A generalized additive model is of the form: $g(E(Y|x)) = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_m(x_m)$

It is a generalization of generalized linear model (GLM), which is given by: $g(E(Y|x)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m$ GAMs do not allow for interaction terms. Use $E(f_j(x_j))$ for identifiability

3 Tree-based Classifiers

[*nonparametric*] Find partitions that minimize RSS: $\sum_{j=1}^{J}\sum_{i\in R_j}(y_i - \hat{y}_{R_j})^2$, ($R_j$ partitions, $\hat{y}_{R_j}$ mean response)
Variable Importance: total amount RSS decreased due to predictor, averaged over all trees. Trees can model interactions between predictors. Computationally infeasible to consider all divisions

3.1 Regression Trees

Gini index: $\sum_{k=1}^{K} p_{mk}(1-p_{mk})$ measures total variance across K classes; small value $\implies$ node has predominantly one class.
Cross Entropy: $-\sum_{k=1}^{K} p_{mk}\log p_{mk}$

3.2 Tree Pruning

Weakest Link Pruning minimizes $\sum_{m=1}^{|T|}\sum_{x_i\in R_m}(y_i - \hat{y}_{R_m})^2 + \alpha|T|$

3.3 Bagging

Definition 3.1 (Out-of-Bag Error)**.** Use remaining ∼1/3 observations to calculate error. If B large, this is leave-one-out CV error (B: # of bs samples). $\text{Err}_{OOB} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_{i,OOB})$

3.4 Random Forest

Decorrelates the trees. Each split has random selection of m (of full set p) predictors. Typically, $m \approx \sqrt{p}$. Cannot overfit. Adding more trees cannot hurt you.

3.5 Boosting

Unlike fitting large decision tree (and potentially overfit), boosting learns slowly. *Tuning Parameters* B: # of trees; λ: shrinkage param, learning rate, typically 0.01 or 0.001; # of splits: depth of tree

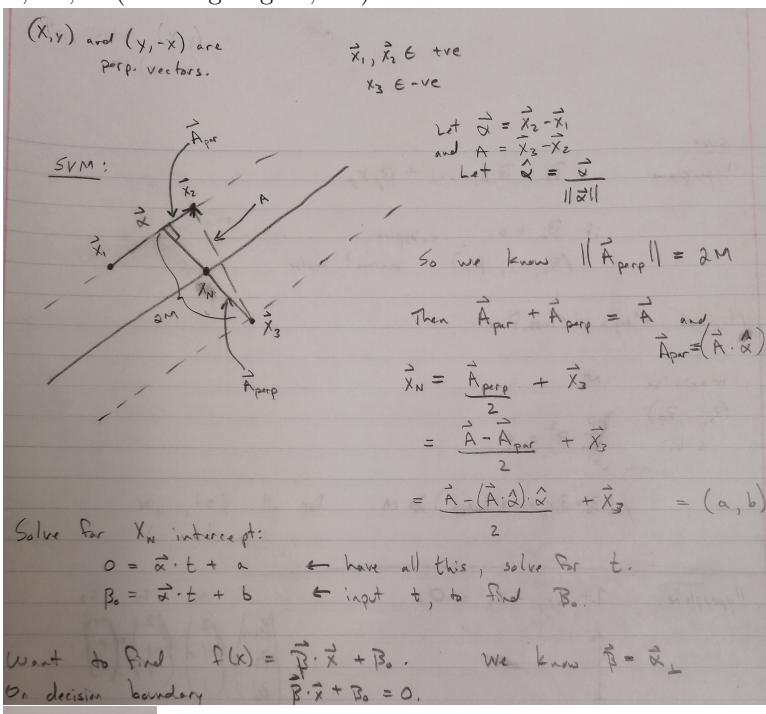
3.5.1 AdaBoost

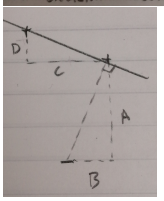
(1) Forest of stumps, (2) some stumps weighted heavier, by α_m, (3) built sequentially. Weight misclassified obs by factor $\alpha_m = \log\left(\frac{1-\text{err}_m}{\text{err}_m}\right)$ where $\hat{G}_m(x)$ classifer & $\text{err}_m = \frac{\sum_{i=1}^{n} w_{mi} I(y_i \neq \hat{G}_m(x_i))}{\sum_i w_{mi}}$. Stopping criteria hard to determine. AdaBoost is equivalent to forward stepwise additive with loss as $L(y, f(x)) = e^{-yf(x)}$
Stat. Property: $f^*(x) = \text{argmin}_f E\left[e^{-Yf(x)}\right] = \frac{1}{2}\log\frac{P(Y=1|x)}{P(Y=-1|x)}$
Proof: $E(e^{-Yf(x)}) = P(Y=1|x)e^{-f(x)} + P(Y=-1|x)e^{f(x)} = \frac{P(Y=1|x)}{P(Y=-1|x)}e^{-f(x)} + e^{f(x)}$. Take deriv. w.r.t. $f(x)$, set to zero: $0 = -\frac{P(Y=1|x)}{P(Y=-1|x)}e^{-f(x)} + e^{f(x)} \implies f(x) = \frac{1}{2}\log\frac{P(Y=1|x)}{P(Y=-1|x)}$ **Gradient Boosting** *Tuning Param*: # iters M; tree size J; shrinkage param ν; **XGBoost**: Optimized for efficiently reduce computing time and allocate an optimal usage of memory resources

4 Support Vector Machines

[*discriminative/parametric*] A hyperplane: $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$. If $\beta_0 = 0$, a subspace. $(\beta_1, \beta_2, \ldots, \beta_p)$: normal vector. When classes nearly separable, SVM better than LR, also LDA. For non-separable, using budget $C = 0$ implies no budget for violations.

$y_i(\omega x_i + b) \geq 1$ or $y_i\left((\omega\ b)\cdot\begin{pmatrix}x\\1\end{pmatrix}\right) \geq 1$. Hyperplane: $\omega = \omega_1 x_1 \cdots + \omega_n x_n$. Width of street: $M = \frac{1}{\|\omega\|}$ (without intercept). To maximize M, $\min \frac{1}{2}\|\omega\|^2$ s.t. $y_i(x_i\omega + b) \geq 1$ for $i = 1, \ldots, n$. (find Lagrangian, ...)





If $\frac{|D|}{|C|} > \frac{|B|}{|A|}$ then use two points.

4.1 Kernel SVMs

4.1.1 Radial Kernel

[*nonparametric*] $K(x_i, x_{i'}) = \exp\left(-\gamma\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2\right)$, $\gamma > 0$. It is nonparametric because pairwise distances are calculated between training points, γ is determined by CV. It is a basis expansion, basically the relationship btw 2 pts in ∞ dimensions

4.1.2 d-degree Polynomial Kernel

$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij}x_{i'j})^d$. d is usually determined using CV. A basis expansion in higher dimensions.

4.2 More than 2 classes

4.2.1 One-versus-One

Constructs $\binom{K}{2}$ SVMs. Tally # of times the obs is assigned to each of K classes. Most frequent class wins

4.2.2 One-versus-All

Construct K SVMs, to get $\beta_{0k}, \beta_{1k}, \ldots, \beta_{pk}$ for each k. Let x^* be test obs. Classify to k, where $\beta_{0k} + \beta_{1k}x_k^* + \cdots + \beta_{pk}x_p^*$ largest (most confident) **Neural Networks**[*discriminative*]