Graphical Representation of Numerical Data

Kernel density estimator:

$$\hat{f}_h(x) = \frac{1}{2nh}\sum_{i=1}^{n} I(|x - x_i| \leq h) \quad \text{(Histogram)}$$

$$\Rightarrow \hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

with Kernel $K(u) = I(|u| \leq 0.5)$ (uniform Kernel), h...Degree of smoothness

Decision Trees

Impurity functions $u(R)$: $p_k(R) = \frac{1}{n}\sum I(y_i = k)$ (purity)

- classification error: $1 - \max_k p_k(R)$
- Gini index: $1 - \sum_k p_k(R)^2$
- entropy: $-\sum_k p_k(R) \cdot \log(p_k(R))$

⇒ are all maximized when p_k is uniform on the K classes in R; all are minimized when $p_k = 1$ for some k (R has one class)

Growing a classification tree: $R \to R^+$ and R^-

- calculate $u(R)$, $u(R^+)$, $u(R^-)$
- Gini improvement: $u(R) - (p(R^-) \cdot u(R^-) + p(R^+) \cdot u(R^+)) \to \max$

⇒ reduces uncertainty

Growing a regression tree:

$$\hat{y} = \sum_{i=1}^{M} c_m I(x \in R_m)$$

$$\Rightarrow \hat{c}(R_m) = \frac{1}{n(R_m)}\sum_{i=1}^{n} I(y_i \mid x_i \in R_m)$$

Stopping parameters:

- #splits, #observations per region, Δ objective function
- tree prunning: $R_\alpha(T) = \frac{1}{\sum(y_i - \bar{y})^2}\sum_{m=1}^{|T|}\sum_{i:x_i \in R_m}(y_i - f(x_i))^2 + \alpha|T| \to \min$
- α...complexity parameter (CP)

Optimal subtree with CV:

- trees T_0 (0 splits), ..., T_m (m splits) ⇒ $\infty, \alpha_1, ..., \alpha_{min}$
- $\beta_i = \sqrt{\alpha_i \cdot \alpha_{i+1}}$ (average)
- subsets $G_1, ..., G_B$
- trees with $\beta_1, ..., \beta_m$ for each subset (leave one out → forecast)

⇒ smallest β over all G_i's

Bagging: bootstrapping from training data, tree T_i

- prediction: $\frac{1}{n}\sum_{i=1}^{n} \text{pred}(T_i, x)$
- when correlation in bootstrap samples → effect decreases

Random Forests: bootstrapping + subset of explanatory variables

- importance of variable: how much increase of MSE or classification error when variable is permuted in left-out-sample

Boosting: AdaBoost:

- bootstrapping from training data with distribution w_t ($w_0 = \frac{1}{n}$)
- train classifier f_t
- $\varepsilon_t = \sum w_t \cdot I(y_t \neq f_t(x_i))$, $\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
- scale $w_{t+1}(i) = w_t(i) \cdot \exp(-\alpha_t y_i f(x_i))$ and normalize
- training error $\frac{1}{n}\sum I(y_i \neq f_{boost}(x_i)) \leq \exp\left(-2\sum\left(\frac{1}{2} - \varepsilon_t\right)^2\right)$

⇒ $f_{boost} = \text{sgn}\left(\sum \alpha_t f_t\right)$

CHAID: not binary, split so that ANOVA has smallest p-value

Nearest Neighbors Classifiers

NN classifier: label x with label of closest point (default euclidean distance)

k-NN classifier: majority vote of k closest points

Linear classifier and Perceptron

Hyperplane H: $p - 1$ dimensional subspace $H = \{x \mid \langle x, w\rangle = 0\}$

- affine Hyperplane $H = \{x \mid \langle x, w\rangle + w_0 = 0\}$

⇒ linear classifier: $\text{sgn}(\langle x, w\rangle + w_0)$

Perceptron:

- $L = -\sum(y_i \cdot \langle x_i, w\rangle) I(y_i \neq \text{sgn}(\langle x, w\rangle)) \to \min$
- $\frac{\partial L}{\partial w}$ direction in which L is increasing ⇒ $w' = w - \eta\frac{\partial L}{\partial w}$ (gradient descent, Perceptron uses a stochastic version): find one $(x_i y_i)$ where $y_i \neq \text{sgn}(\langle x_i, w\rangle)$ and then $w_{t+1} = w_t + \eta y_i x_i$
- problems: data separable → one H will be found (not necessary the best), data not separable → algorithm won't stop

Maximum Margin Classifiers and SVM

Maximum margin classifier:

- margin: distance $H \leftrightarrow$ closest point → max
- convex hull: every point can be reached by linear combination of convex-hull-points
- How to find H?

→ $\langle x_i, w\rangle + w_0 \geq 1$ ($y_i = 1$), $\langle x_i, w\rangle + w_0 \leq -1$ ($y_i = -1$) ⇒ $y_i(\langle x_i, w\rangle + w_0) - 1 \geq 0$

→ $d_+ = d_- = \frac{1}{\|w\|} \Rightarrow d_+ + d_- = \frac{2}{\|w\|} \to \max$

→ primal problem: $L = \frac{1}{2}\|w\|^2 \to \min$ s.t. $y_i(\langle x_i, w\rangle + w_0) \geq 1$ (with Lagrange coefficients α)

→ dual problem $L = \sum \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle \to \max$ s.t. $\sum \alpha_i y_i = 0$ (find closest points in convex hull)

→ solve dual problem for α, $w = \sum \alpha_i y_i x_i$, pick i for $\alpha_i > 0$ and solve $y_i(\langle x_i, w\rangle + w_0) \geq 1$ for w_0

→ problems: robustness, if not linear separable → not working

Support Vector Classifier:

- slack variables ξ_i represent violation from strict separation, $y_i(\langle x_i, w\rangle + w_0) \geq 1 - \xi_i$
- $L = \frac{1}{2}\|w\|^2 + \lambda\sum\xi_i$ s.t. $y_i(...) \geq 1 - \xi_i$
- problems: non separability

Support Vector Machine: maps data in higher dimensions with $\phi(x)$

- primal problem: SVC with $x \to \phi(x)$
- dual problem: MMC with $x \to \phi(x)$ s.t. $0 \leq \alpha_i \leq \lambda$, $\sum \alpha_i y_i = 0$

⇒ if K exists such that $K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$ we can use SVM without knowing ϕ (kernel trick)

- $K(x_i, x_j)$ pos. def. $\Leftrightarrow \sum\sum \lambda_i\lambda_j K(x_i, x_j) > 0$

→ linear kernel ($\langle x_i, x_j\rangle$), polynomial kernel ($(\langle x_i, x_j\rangle + 1)^p$), radial kernel ($\exp(-\|x_i - x_j\|/2\sigma^2)$), sigmoid kernel ($\tanh(k\langle x_i, x_j\rangle - \delta)$)

- SVM with K classes: 1-vs-1: $\binom{K}{2}$ pairs, $\binom{K}{2}$ classifiers ⇒ majority vote; 1-vs-all: compare one of K classes to rest, assign x^* to $b_k + w_{1k}x_1^* + ... \to \max$