Parzen - Rosenblatt Estimator

- Nonparametric Density Estimation

Kernel window Function:

$$k(\vec{x}_i, \vec{x}) = \begin{cases} 1, & \frac{|x_{i,k} - x_k|}{h} \leq 1/2 \\ 0, & \text{otherwise} \end{cases}$$

(Box Kernel)

$$p_R = p(\vec{x}) \cdot V_R$$

$$= \frac{k_R}{N} \cdot V_R$$

$$\Rightarrow p(\vec{x}) = \frac{k_R}{N \cdot V_R} \quad \rightarrow \text{use box kernel} \rightarrow \boxed{p(\vec{x}) = \frac{1}{N \cdot h^d} \cdot \sum_{i=1}^{N} k_h(\vec{x}_i, \vec{x})}$$

Cross Validation

$$\hat{h} = \underset{h}{\operatorname{argmax}}\ \alpha(h) \overset{ML}{=} \underset{h}{\operatorname{argmax}} \prod_{j=1}^{N} p^j_{h,N-1}(\vec{x}_j)$$

$$= \boxed{\underset{h}{\operatorname{argmax}} \sum_{j=1}^{N} \log\left[p^j_{h,N-1}(\vec{x}_j)\right]}$$

$S = \{x_1, \ldots, x_N\}$

$S^j = S \setminus \{x_j\}$ ← test sample

Mean shift Algorithm

1. Compute Mean shift Vector:

$$\frac{\sum_{i=1}^{N} k'_h(\|x_i - x\|^2) \cdot x_i}{\sum_{i=1}^{N} k'_h(\|x_i - x\|^2)} - x = \vec{\varnothing}$$

2. Update x:

$$x^{(t+1)} = x^{(t)} + m(x^{(t)})$$

$$= \frac{\sum_{i=1}^{N} k'_h(\|x_i^{(t)} - x^{(t)}\|^2) \cdot x_i^{(t)}}{\sum_{i=1}^{N} k'_h(\|x_i^{(t)} - x^{(t)}\|^2)} + x^{(t)} - x^{(t)}$$

Epanechnikov Kernel:

$$k_E(x) = \begin{cases} c \cdot (1 - x^T x), & x^T x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

PA Summary (2) — Clustering — 24.07.13

K - Means

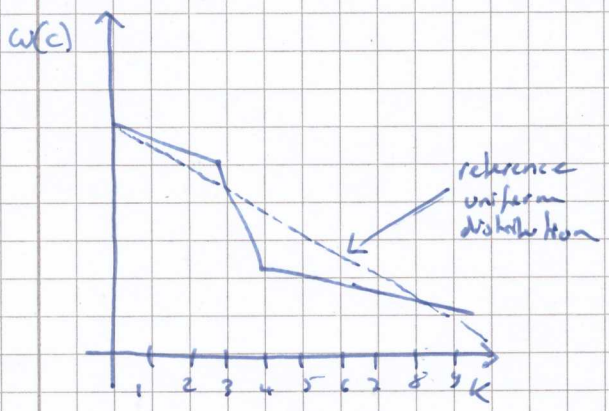
1. For a given cluster assignment C: compute μ
2. Assign each observation to the closest μ
3. Iterate 1. and 2. until convergence

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

Using medoids increases

$O(N) \rightarrow O(N^2)$ (additional pair checking)

K model selection problem



Tibshirani Gap statistics

Tibshirani Significance:

1.) Repeat sampling B times

2.) Calculate standard deviation of $W(C)$ for each K of the synthetic curve

$$K^* = \underset{K}{\operatorname{argmin}} \left\{ K \mid G(K) \geq G(K+1) - s'_{K+1} \right\}$$

$$s'_{K+1} = E\left(\log(W_K) - \log(W_K)\right)$$

Hierarchical clustering

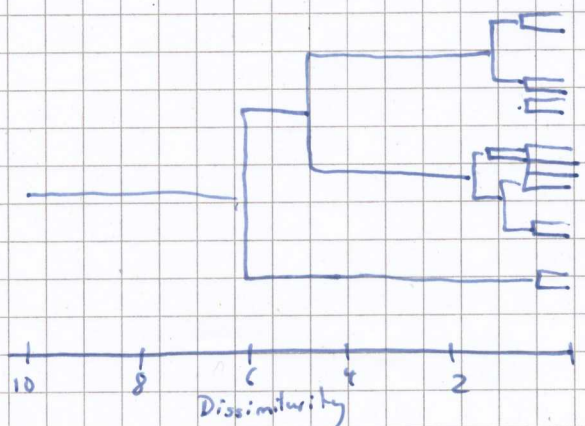
- top-down: "divisive clustring"
- bottom-up: "agglomerative clustering"

Single Linkage: Merge least dissimilar points
-> chains may form

Complete Linkage: merge where largest sample distance is small

Group Average: merge where avg distance is small

Dendrogram:



Model Selection for GMMs

$$p(\vec{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)$$

- Dirichlet Process Mixture
- Prior distribution of GMM Parameters

$$p_{ik} \equiv \gamma(z_{nk})$$
$$p_k = \pi_k$$

GMM Expectation Maximization: (classic)

0.) Initialize π_k, $\vec{\mu}_k$, Σ_k

1.) "Expectation":

$$\gamma(z_{nk}) = \frac{\pi_k \cdot \mathcal{N}(\vec{x}_n \mid \mu_k, \Sigma_k)}{p(\vec{x}_n)} \quad , \quad p(\vec{x}_n) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}_n \mid \mu_j, \Sigma_j)$$

weighted average

$$\sum_K \quad \textcircled{x}^{10\%} \quad \textcircled{x}^{?} \quad \textcircled{x}^{100\%} \quad \textcircled{x}^{50\%}$$

2.) "Maximization":

$$\vec{\mu}_k^{new} = \frac{1}{N_k} \cdot \sum_{n=1}^{N} \gamma(z_{nk}) \cdot \vec{x}_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \cdot \sum_{n=1}^{N} \gamma(z_{nk}) \cdot (\vec{x} - \vec{\mu}_k^{new}) \cdot (\vec{x} - \vec{\mu}_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N} \quad , \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Gibbs sampler:

0.) Assign initial clusters

1.) Randomly choose sample $\vec{x}_i$, remove from clustering

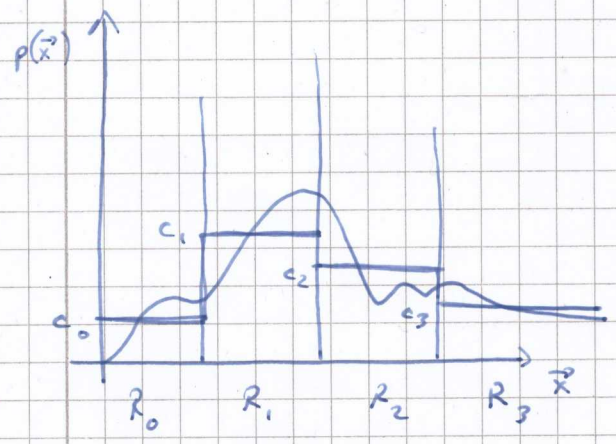
2.) Assignment to a new seat:

(for existing cluster) $p(z_i = k) \cdot p(\vec{x}_i \mid \mu_k, \Sigma_k) = \frac{N_k}{N + \alpha_0} \cdot \mathcal{N}(\vec{x}_i \mid \vec{\mu}_k, \Sigma_k)$

"rich get richer" — "do I like the food?"

(for a new cluster) $\frac{\alpha_0}{N + \alpha_0} \cdot \mathcal{N}(\vec{x}_i \mid \mu_0, \Sigma_0)$

Regression Tree



$R_j \hat{=}$ leaf node j

$$\hat{c}_j = \underset{c_j}{\operatorname{argmin}} \int_{R_j} \left(p(\vec{x}) - \hat{p}(\vec{x})\right)^2 d\vec{x}$$

$$= \underset{c_j}{\operatorname{argmin}} \int_{R_j} \left(p(\vec{x}) - \sum_{j=1}^{k} c_j \, I(\vec{x} \in R_j)\right)^2 d\vec{x}$$

$$c_j = \frac{\int_{R_j} p(\vec{x})\, dx}{\int_{R_j} 1\, dx}$$

$\Rightarrow c_j$ is the mean of the values in R_j

Mitigate overfitting

Bagging: Train many trees on random subsets of data

Boosting: Train a sequence of classifiers with increasing difficulty. Make a weighted vote (AdaBoost, Viola & Jones)

Random Forests: Randomize Training Process:
- random data subset
- random dimension for split
- finite subset of parameters for split function

Entropy:

Classification:

$$H(S) = -\sum_{c \in \mathcal{C}} p(\vec{x} \in c) \cdot \log\left(p(\vec{x} \in c)\right)$$

$\mathcal{C}$: all class labels

c: class label

Regression:

$$H(S) = -\frac{1}{|S|} \sum_{x \in S} \int_{y} p(y|x) \cdot \log\left(p(y|x)\right) dy$$

$$p(y|x) = \mathcal{N}(y;\ \bar{y}(x),\ \bar{\sigma}_y^2(x))$$