Unsupervised Learning

- No goal class (either Y nor G).
- We are interested in relations in the data:

  Clustering Are the data organized in natural clusters? (Clustering, Segmentation)
  k-means
  hierarchical clustering
  EM algorithm for clustering
  (Dirichlet Process Mixture Models)
  (Spectral Clustering)

  Association Rules Are there some frequent combinations, implication relations? (Market Basket Analysis) *later*

  Other The Elements of Statistical Learning Chapter 14

  SOM Self Organizing Maps

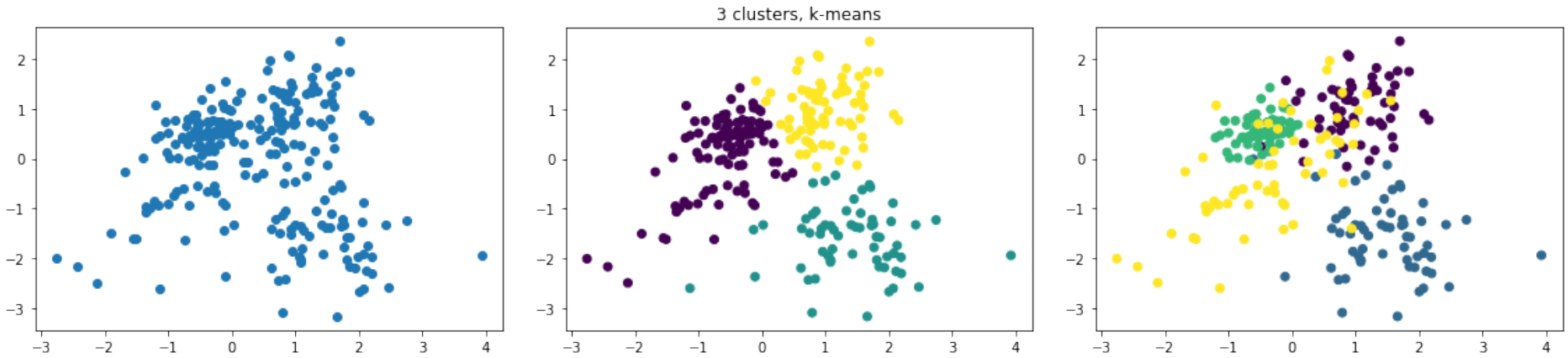
  PCA Principal Component Analysis Linear Algebra; k linear combinations of features minimizing reconstruction error (= first k principal components).

  - Principal Curves and Surfaces, Kernel and Spare Principal Components

  ICA Independent Component Analysis.

Clustering Example

- We have unlabeled train data.
- We want to assign same cluster/color to nearby points.
- You may not be able to recover the true data origin if the mixture components overlap.



K – means !

K–means

1: **procedure** $K-\text{MEANS}$:(X data, K the number of clusters)
2: select randomly K centers of clusters μ_k
3: # either random data points or random points in the feature space
4: **repeat**
5: **for** each data record **do**
6: $C(x_i) \leftarrow argmin_{k \in \{1,\ldots,K\}} d(x_i, \mu_k)$
7: **end for**
8: **for** each cluster k **do** # find new centers μ_k
9: $\mu_k = \sum_{x_i : C(x_i)=k} \frac{x_i}{|C(k)|}$.
10: **end for**
11: **until** no chance in assignment
12: **end procedure**

K – means

K–means

The t iterations of K–means algorithm take $O(tKpN)$ time.

- To find global optimum is NP-hard.
- The result depends on initial values.
- May get stuck in local minimum.
- May not be robust to data sampling.
  - We may generate datasets by bootstrap method.
  - The cluster centers found in different dataset may be quite different.

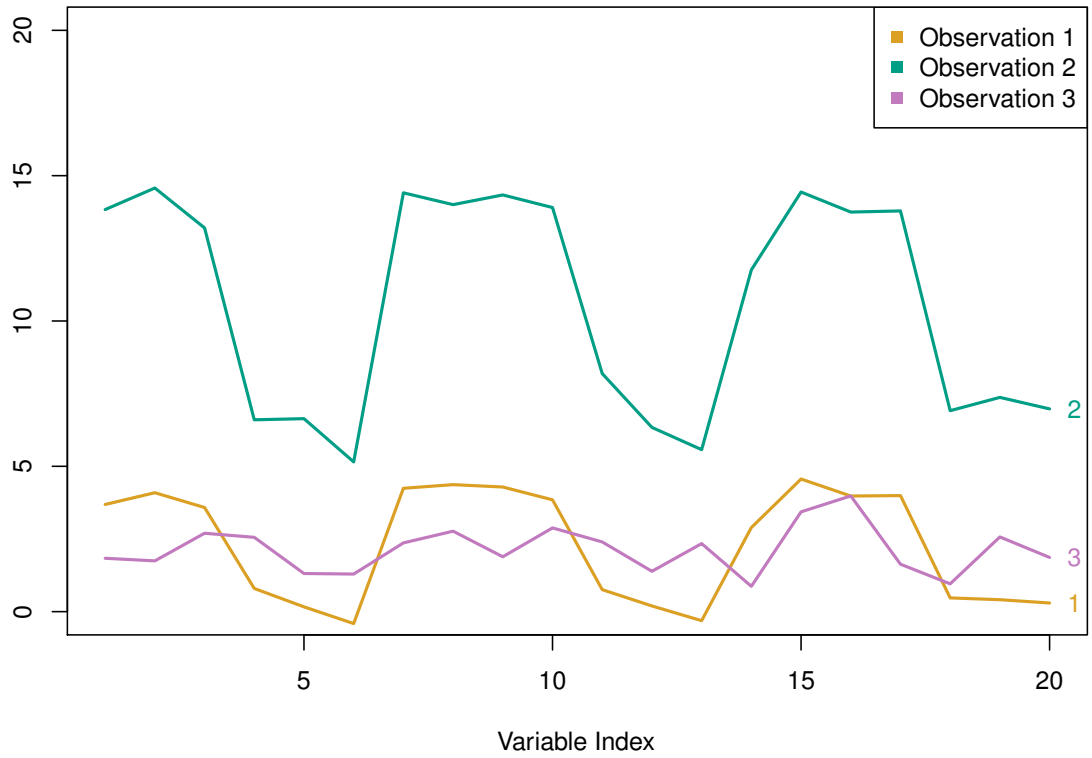
  (for example, different bootstrap samples may give very different clustering results).
- Each record must belong to some cluster. Sensitive to outliers.

Distance measures

the most common distance measures:

Euclidian	$d(x_i, x_j) = \sqrt{\sum_{r=1}^{p}(x_{ir} - x_{jr})^2}$
Hamming (Manhattan)	$d(x_i, x_j) = \sum_{r=1}^{p} \lvert x_{ir} - x_{jr} \rvert$
overlap (překrytí) categorical variables	$d(x_i, x_j) = \sum_{r=1}^{p} I(x_{ir} \neq x_{jr})$
cosine similarity	$s(x_i, x_j) = \frac{\sum_{r=1}^{p}(x_{ir} \cdot x_{jr})}{\sqrt{\sum_{r=1}^{p}(x_{jr} \cdot x_{jr}) \cdot \sum_{r=1}^{p}(x_{ir} \cdot x_{ir})}}$
cosine distance	$d(x_i, x_j) = 1 - \frac{\sum_{r=1}^{p}(x_{ir} \cdot x_{jr})}{\sqrt{\sum_{r=1}^{p}(x_{jr} \cdot x_{jr}) \cdot \sum_{r=1}^{p}(x_{ir} \cdot x_{ir})}}$

Other Distance Measures



Correlation Proximity

- Euclidian distance: Observations 1 and 3 are close.
- Correlation distance: 1 and 2 look very similar.

$$\rho_{X,Y} = corr(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

Distance – key issue, application dependent

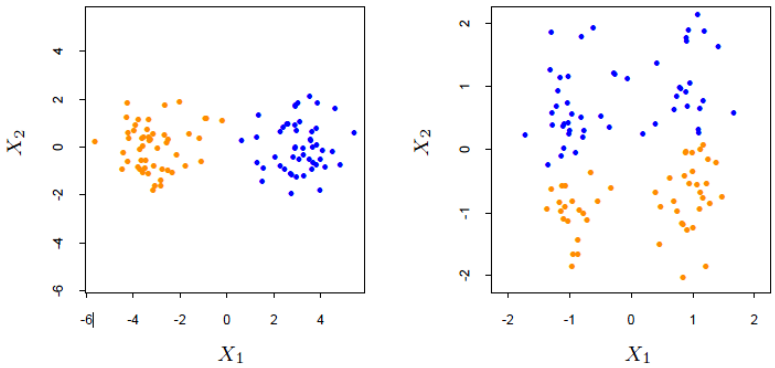
- The result depends on the choice of distance measure $d(x_i, \mu_k)$.
- The choice is application dependent.
- Scaling of the data is recommended.
- Weights for equally important attributes j are: $w_j = \frac{1}{\hat{d}_j}$ where

$$\hat{d}_j = \frac{1}{N^2} \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} d_j(x_{i_1}, x_{i_2}) = \frac{1}{N^2} \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} (x_{i_1}[j] - x_{i_2}[j])^2$$

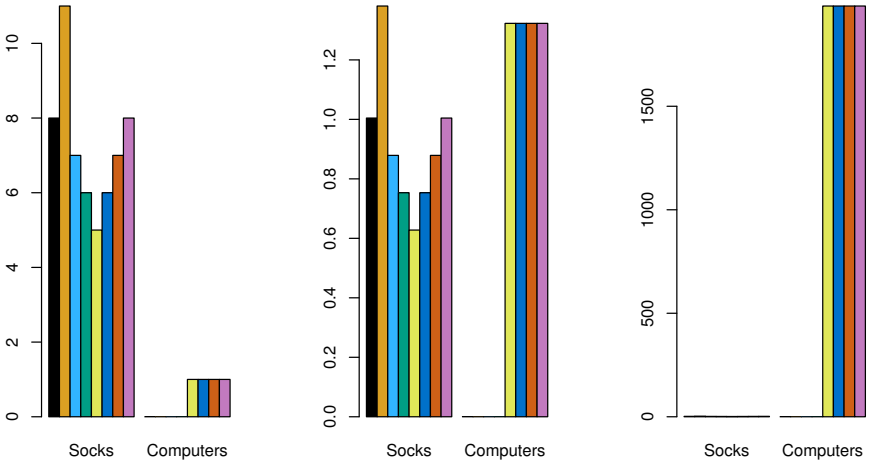
- Total distance as a weighted sum of attribute distances.
- Distance may be specified directly by a symmetric matrix, 0 at the diagonal, should fulfill triangle inequality

$$d(x_i, x_\ell) \leq d(x_i, x_r) + d(x_r, x_\ell).$$

Alternative Ideas



- Scaling may remove natural clusters
- Weighting Attributes
  - Consider internet shop offering socks and computers.
  - Compare: number of sales, standardized data, $



Number of Clusters

- We may focus on the **Within cluster variation** measure:

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

- Notice that $W(C)$ is decreasing also for uniformly distributed data.
- We look for small drop of $W(C)$ as a function of K or maximal difference between $W(C)$ on our data and on the uniform data.
- **Total** cluster variation is the sum of **between** cluster variation and **within** cluster variation

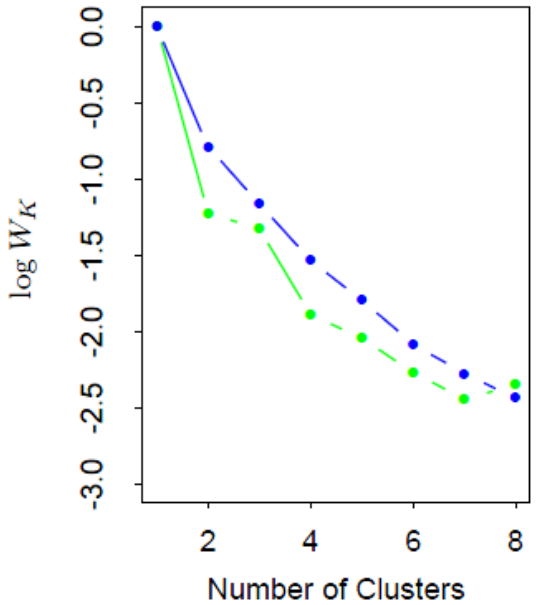
$$\begin{aligned} T(C) &= \frac{1}{2}\sum_{i,i'=1}^{N} d(x_i, x_{i'}) = W(C) + B(C) \\ &= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\Big(\sum_{C(i')=k} d(x_i, x_{i'})\Big) + \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\Big(\sum_{C(i')\neq k} d(x_i, x_{i'})\Big) \end{aligned}$$

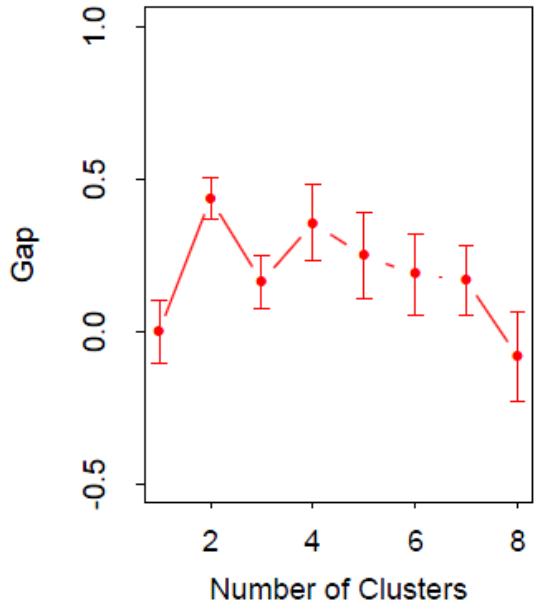
GAP function for Number of Clusters

- denote W_k the expected W for uniformly distributed data and k clusters, the average over 20 runs
- GAP is expected $log(W_k)$ minus observed $log(W(k))$

$$K^* = argmin\{k|G(k) \geq G(k+1) - s'_{k+1}\}$$

$$s'_k = s_k\sqrt{1+\frac{1}{20}} \text{ where } s_k \text{ is the standard deviation of } log(W_k)$$





Silhouette

For each data sample x_i we define

- $a(i) = \frac{1}{|C_i|-1} \sum_{j \in C_i, i \neq j} d(i,j)$ if $|C_i| > 1$
- $b(i) = min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$

Optimal number of clusters k may be selected by the SC.

Definition (Silhouette)

Silhouette s is defined

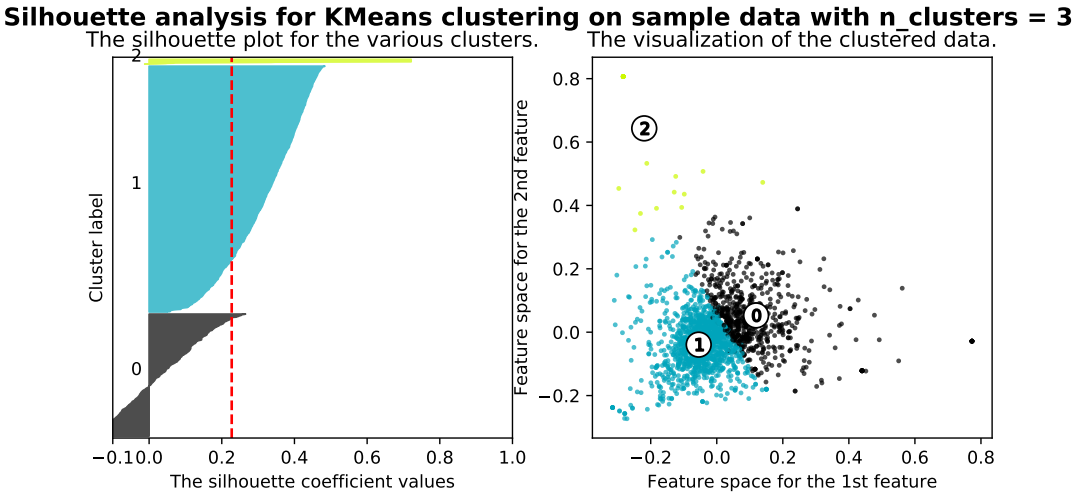
- $s(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}}$ if $|C_i| > 1$
- $s(i) = 0$ for $|C_i| = 1$.

Definition (Silhouette Score)

The Silhouette score is $\frac{1}{N} \sum_i^N s(i)$.

Silhouette is always between

- $-1 \leq s(i) \leq 1$.



Note: One cluster $(-1,1),(1,1)$, other cluster $(0,-1.2),(0,-1.1)$, the point $(0,0)$ is assigned to the first cluster but has a negative silhouette. https://stackoverflow.com/a/66751204

Country Similarity Example

- Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.

	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS	YUG
BRA	5.58										
CHI	7.00	6.50									
CUB	7.08	7.00	3.83								
EGY	4.83	5.08	8.17	5.83							
FRA	2.17	5.75	6.67	6.92	4.92						
IND	6.42	5.00	5.58	6.00	4.67	6.42					
ISR	3.42	5.50	6.42	6.42	5.00	3.92	6.17				
USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17		
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92

K–medoids

1: **procedure** $K-$MEDOIDS:(X data, K the number of clusters)
2: select randomly K data samples to be centroids of clusters
3: **repeat**
4: **for** each data record **do**
5: assign to the closest cluster
6: **end for**
7: **for** each cluster k **do** # find new centroids $i_k^* \in C_k$
8: $i_k^* \leftarrow argmin_{\{i:C(i)=k\}} \sum_{C(i')=k} d(x_i, x_{i'})$
9: **end for**
10: **until** no chance in assignment
11: **end procedure**

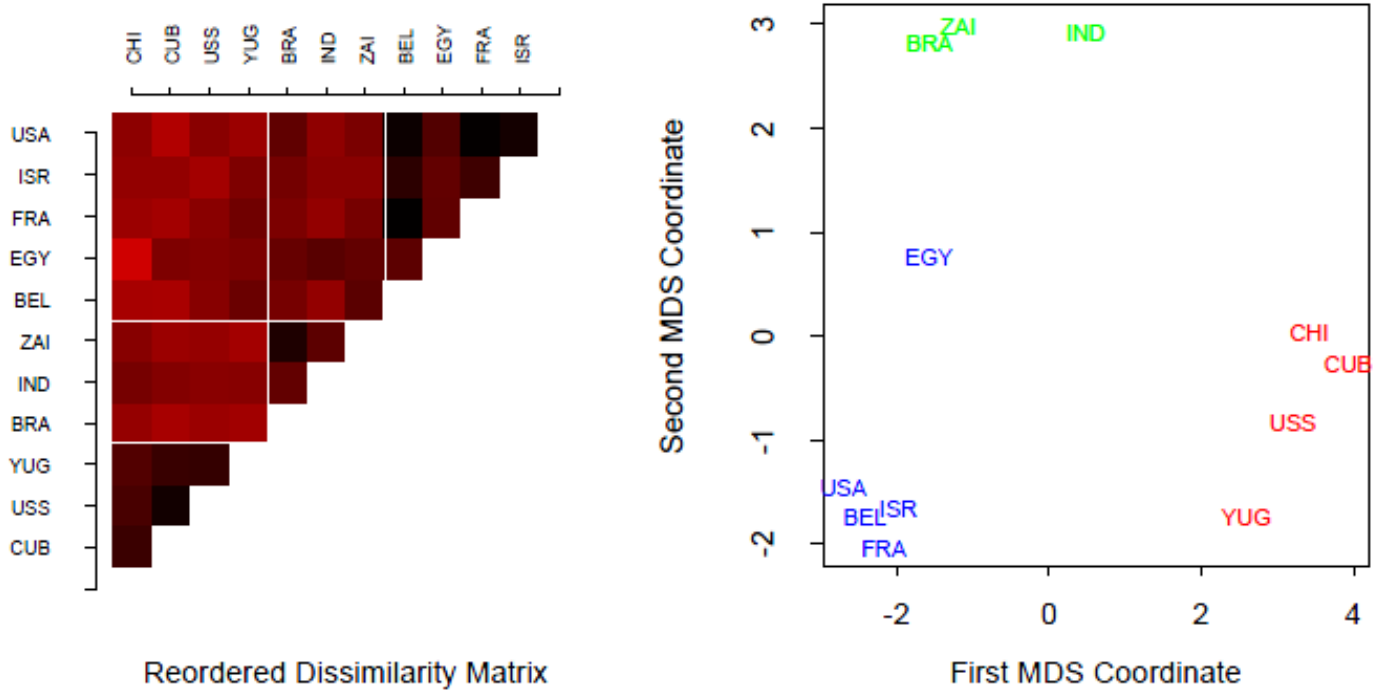
- To find a centroid requires quadratic time compared to linear k–means.
- We may use any distance, for example number of differences in binary attributes.

Complexity

The t iterations of K–medoids take $O(tkpN^2)$.

Clusters of Countries

- Survey of country dissimilarities.
- Left: dissimilarities
  - Reordered and blocked according to 3-medoid clustering.
  - Heat map is coded from most similar (dark red) to least similar (bright red).
- Right: Two-dimensional multidimensional scaling plot
  - with 3-medoid clusters indicated by different colors.



Multidimensional Scaling

- The right figure on previous slide was done by Multidimesional scaling.
- We know only distances of countries, not a metric space.
- We try to keep proximity of countries (*least squares scaling*).
- We choose the number of dimensions p.

Definition (Multidimensional Scaling)

For a given data $x_1, \ldots, x_N$ with their distance matrix d, we search $(z_1, \ldots, z_N) \in \mathbb{R}^p$ projections of data minimizing stress function

$$S_D(z_1, \ldots, z_N) = \left[\sum_{i \neq \ell} (d[x_i, x_\ell] - \|z_i - z_\ell\|)^2 \right]^{\frac{1}{2}}.$$

- It is evaluated gradiently.
- Note: Spectral clustering.

Hierarchical clustering – Bottom Up

Start with each data sample in its own cluster. Iteratively join two nearest clusters. Measures for join

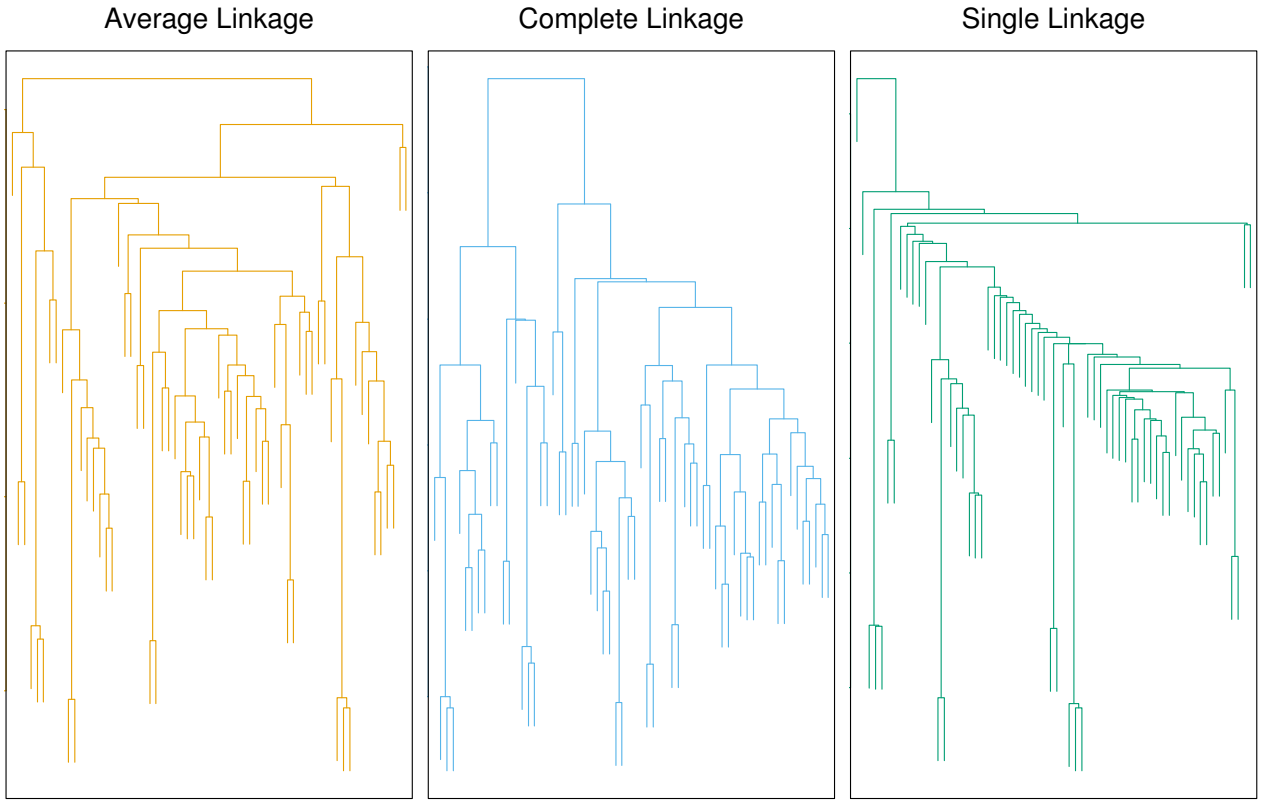
- closest points (**single linkage**)
- maximally distant points (**complete linkage**)
- **average linkage**, $d_{GA}(C_A, C_B) = \frac{1}{|C_A| \cdot |C_B|} \sum_{x_i \in C_A, x_j \in C_B} d(x_i, x_j)$
- **Ward distance** minimizes the sum of squared differences within all clusters.

$$\begin{aligned} Ward(C_A, C_B) &= \sum_{i \in C_A \cup C_B} d(x_i, \mu_{A \cup B})^2 - \sum_{i \in C_A} d(x_i, \mu_A)^2 - \sum_{i \in C_B} d(x_i, \mu_B)^2 \\ &= \frac{|C_A| \cdot |C_B|}{|C_A| + |C_B|} \cdot d(\mu_A, \mu_B)^2 \end{aligned}$$

- where μ are the centers of clusters (A, B and joined cluster).
- It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.

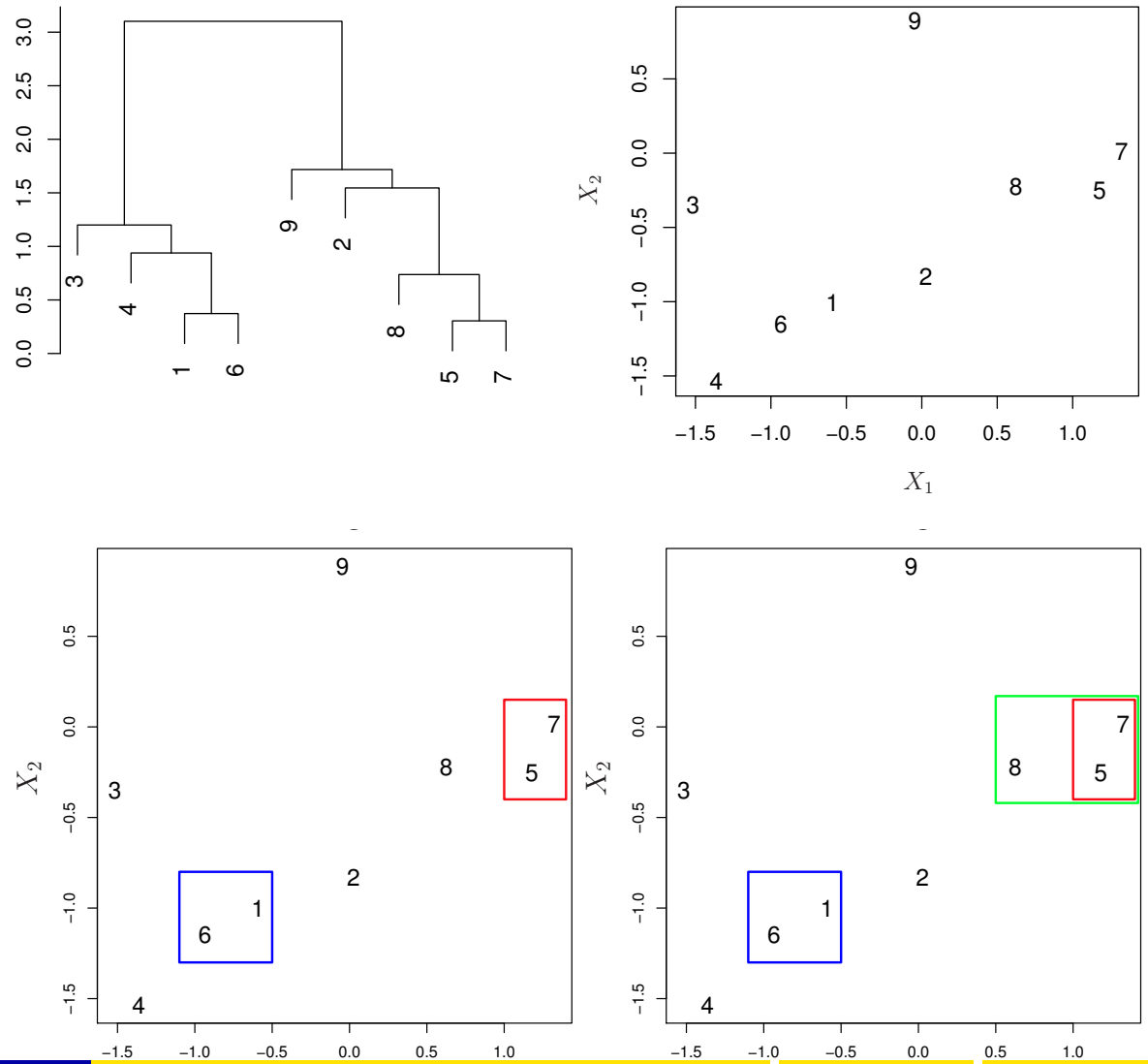
Dendrograms

- Dendrogram is the result plot of a hierarchical clustering.
- Cutting the tree of a fixed high splits samples at leaves into clusters.
  - The length of the two legs of the U-link represents the distance between the child clusters.



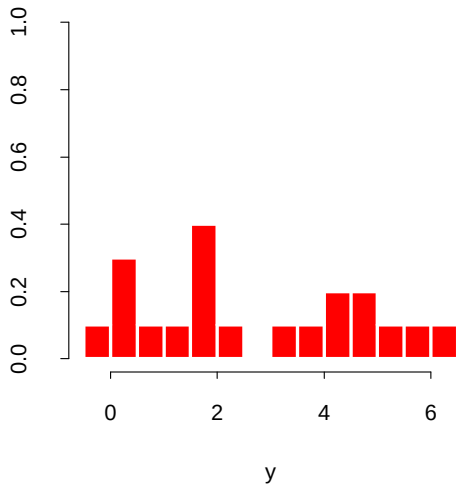
Interpretation of Dendrograms – 2 and 9 are NOT close

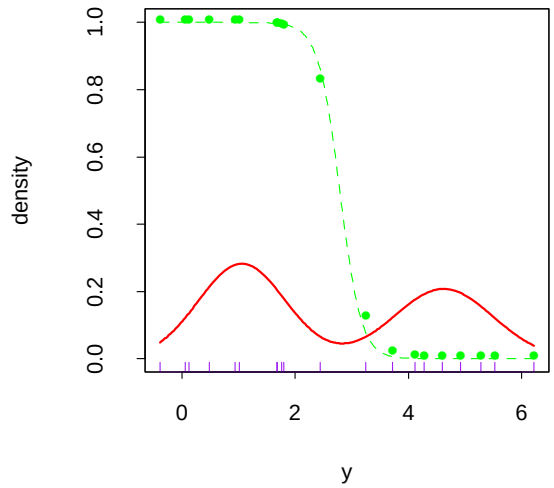
Samples fused at very bottom are close each other.



Gaussian Mixture Model

- Assume the data come from a set of k gaussian distributions
- each with
  - prior probability π_k
  - mean μ_k
  - covariance matrix Σ_k
  - $\phi_{\mu_k,\Sigma_k}(x) = \frac{1}{\sqrt{(2\pi)^p|\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)}$.
- We want to find the maximum likelihood estimate of the model parameters.
- We use (more general) EM algorithm.





EM learning of Mixture of K Gaussians !

- Model parameters $\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k$ such that $\sum_{k=1}^{K} \pi_k = 1$.
- **E**xpectation: weights of unobserved 'fill–ins' k of variable C:

$$\begin{aligned} p_{ik} &= P(C = k|x_i) = \alpha \cdot P(x_i|C_i = k) \cdot P(C_i = k) \\ &= \frac{\pi_k \phi_{\theta_k}(x_i)}{\sum_{l=1}^{K} \pi_l \phi_{\theta_l}(x_i)} \\ p_k &= \sum_{i=1}^{N} p_{ik} \end{aligned}$$

- **M**aximize: mean, variance and cluster 'prior' for each cluster k:

$$\begin{aligned} \mu_k &\leftarrow \sum_i \frac{p_{ik}}{p_k} x_i \\ \Sigma_k &\leftarrow \sum_i \frac{p_{ik}}{p_k} (x_i - \mu_k)(x_i - \mu_k)^T \\ \pi_k &\leftarrow \frac{p_k}{\sum_{l=1}^{K} p_l}. \end{aligned}$$

Dirichlet Process mixture modeling×

- Li Y, Schofield E, Gönen M. A tutorial on Dirichlet Process mixture modeling. J Math Psychol. 2019 https://pmc.ncbi.nlm.nih.gov/articles/PMC6583910/
- How many clusters?
  - Always some probability of a new cluster.
  - Decreases with the current number of clusters.
  - Represented as Dirichlet process (or the Quasi-Bernoulli) Stick-breaking Process
- Finite Mixture Model (fitted by EM algorithm)
  - K clusters with its means μ_k, covariance matrices Σ_k and prior probabilities π_k. Here, $\sum \pi_k = 1$.
  - The likelihood of the model is defined as a mixture of Gaussian distributions

$$p(y_i|\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_k, \pi_1, \ldots, \pi_K) = \sum_{k=1}^{K} \pi_k N(x_i; \mu_k, \Sigma_k))$$

- Dirichlet process prior
  - we introduce the **concentration parameter** α
  - The (prior) probability of a new cluster $K+1$ is $\frac{\alpha}{N-1+\alpha}$
  - The probabilities of a cluster k is: $p(c_i|c_{-i}, \alpha) = \frac{n_{-i,k}}{N-1+\alpha}$ where the assignment c_{-i} of all samples except the i-th is known and $n_{-i,k}$ is the number of samples in the cluster k where the sample i does not count.

Dirichlet Process mixture priors

- We also add prior on the mean of a new cluster $N(\mu_0, \Sigma_0)$
- and a prior on the variance of the new cluster
- actually, on the concentration Σ^{-1} that has the Wishard distribution and its one-dimensional version is denoted by τ.
- and we assume the irreducible noise on x to be known σ_x^2.
- Inside each cluster, the posterior mean is:

$$\mu_k = \frac{\sum_{i \in k} x_i \tau_k + \mu_0 \tau_0}{n_k \tau_k + \tau_0}$$

- the posterior variance is

$$\sigma_k^2 = \frac{1}{n_k \tau_k + \tau_0} + \sigma_x^2.$$

- With large amount of data n_k the prior μ_0 has minimal influence.
- The probability of a sample i being from the cluster k is:

$$p(c_i | c_{-i}, \mu_k, \tau_k, \alpha) = \frac{n_{-i,k}}{N - 1 + \alpha} N\left(x_i; \frac{\sum_{i \in k} x_i \tau_k + \mu_0 \tau_0}{n_k \tau_k + \tau_0}, \frac{1}{n_k \tau_k + \tau_0} + \sigma_x^2\right).$$

MCMC re-assignment

- To make the clustering ordering independent, we iteratively re-assign the samples to a new cluster for predefined number of iterations.

Algorithm 1: DPMM Algorithm

input : $\alpha, \mu_0, \sigma_0^2, \sigma_y^2$ (e.g., $\alpha = 0.01$, $\mu_0 = 0, \sigma_0^2 = I_d \cdot 3^2, \sigma_y^2 = I_d \cdot 1^2$, where I is an identity matrix of $d = 2$), $\tau_0 = 1/\sigma_0^2, \tau_y = 1/\sigma_y^2$.

output: a MCMC chain of simulated values of $\boldsymbol{c}$.

1 Given the concentration parameter α and the state of the Markov chain $\{\mu_k^{(t-1)}, \tau_k^{(t-1)}\}, \boldsymbol{c}^{(t-1)}$, sample a new set of $\{\mu_k^{(t)}, \tau_k^{(t)}\}$ and $\boldsymbol{c}^{(t)}$:

2 **for** $t \leftarrow 1$ **to** *maxiter* **do**

3 **for** $i \leftarrow 1$ **to** n **do**

4 Remove y_i from cluster c_i because we are going to draw a new sample of c_i for y_i.

5 If the previous step causes a cluster to becomes empty, then remove the cluster, its corresponding parameters, and rearrange the order of the clusters into contiguous $1, 2, \ldots, K$.

6 Draw $c_i | c_{-i}, y$ from:

Draw $c_i|c_{-i}, y$ from:

for $k \leftarrow 1$ **to** $K+1$ **do**

Calculate the probability of $c_i = k$ using

$$p(c_i = k|c_{-i}, y_i) \propto \frac{n_{-i,k}}{n-1+\alpha}\mathcal{N}\left(\tilde{y}_i; \frac{\bar{y}_k n_k \tau_k + \mu_0 \tau_0}{n_k \tau_k + \tau_0}, \frac{1}{n_k \tau_k + \tau_0} + \sigma_y^2\right).$$

Calculate the probability of $c_i = k+1$ using

$$p(c_i \neq c_k \ \forall j \neq i|c_i, y_i) \propto \frac{\alpha}{n-1+\alpha}\mathcal{N}(\tilde{y}_i; \mu_0, \sigma_0^2 + \sigma_y^2).$$

if $c_i = k$ for some $j \neq i$ **then**

Update $\bar{y}_k, n_k, \tau_k$ according to $c_i = k$.

else

$c_i = K+1$, a new cluster has been created. Append this new cluster to the vector of non-empty clusters.

end

end

Set $\boldsymbol{c}^{(t)} = c_i$.

end

return $\boldsymbol{c}$

Kernel Density Estimation

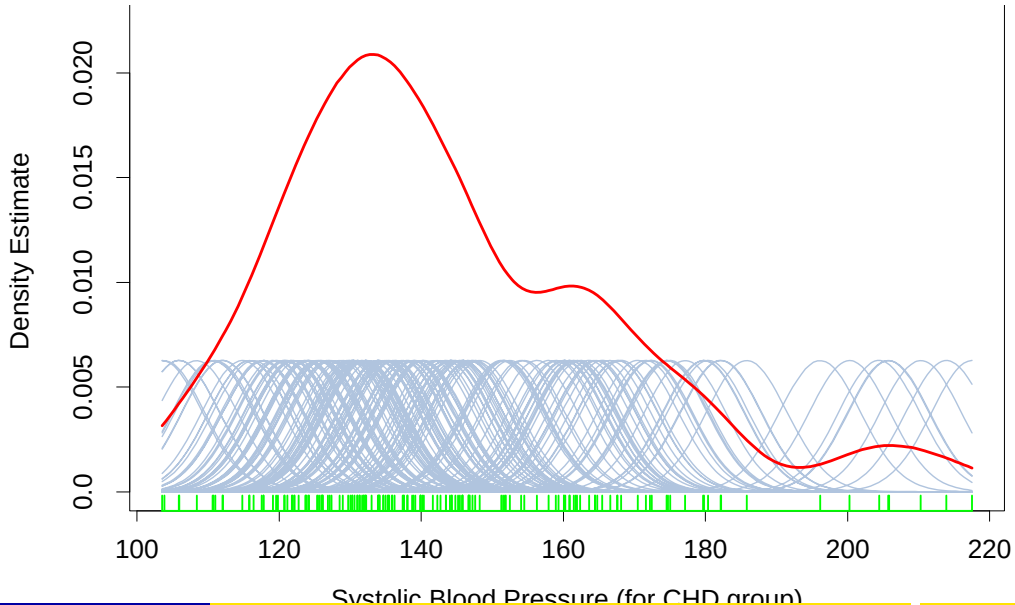
- Kernel Density Estimation is an unsupervised procedure
- We smooth the density estimate in the neighbourhood $\mathcal{N}(x_0)$ with lengthscale λ

$$\hat{f}_X(x_0) = \frac{\#x_i \in \mathcal{N}(x_0)}{N\lambda}$$

- by the Parzen kernel estimate

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

- Popular choice for K_λ is the Gaussian kernel density ϕ_λ.



Kernel Density Classification

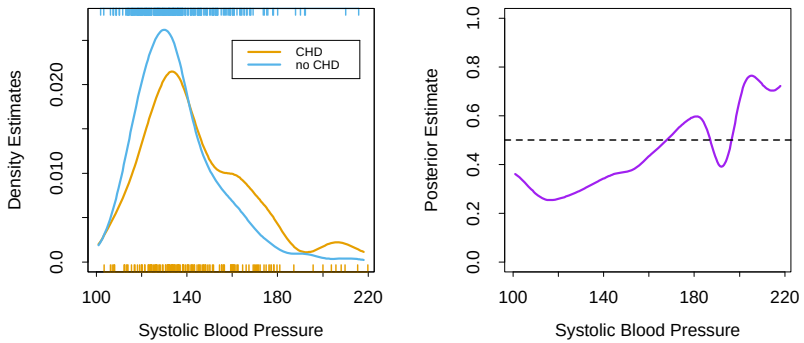
- We may estimate Kernel Density for each target class $k = 1, \ldots, K$, estimate class priors π_k and use Bayes' theorem:

$$\hat{Pr}(G = k|X = x_0) = \frac{\pi_k \hat{f}_k(x_0)}{\sum_{j=1}^{K} \pi_j \hat{f}_k(x_0)}.$$

- by the Parzen kernel estimate

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

- Popular choice for K_λ is the Gaussian kernel density ϕ_λ.



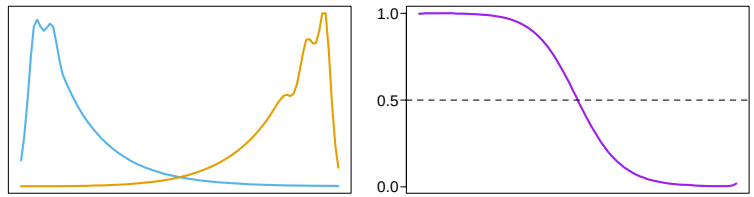


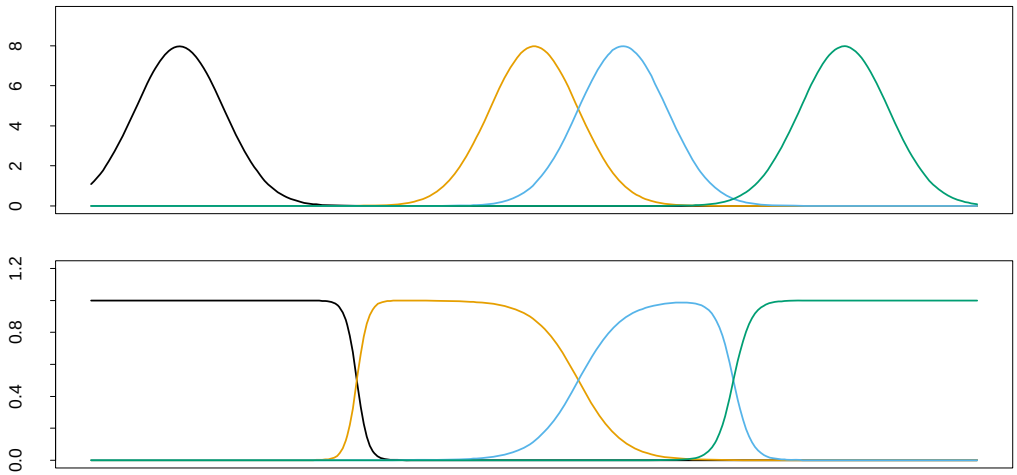
FIGURE 6.15. *The population class densities may have interesting structure (left) that disappears when the posterior probabilities are formed (right).*

Radial Basis Functions and Kernels for Regression

- The kernels do not have to be placed at all observation points.
- We may select (fit) prototype parameters ξ_j and scale patameters λ_j to place pre-defined number of kernels $K_{\lambda_j}(\xi_j, x)$, $j \in 1, \ldots, M, \lambda_j \in \mathbb{R}, \xi_j \in X$.
- and then fit the density as a linear function of kernels as basis

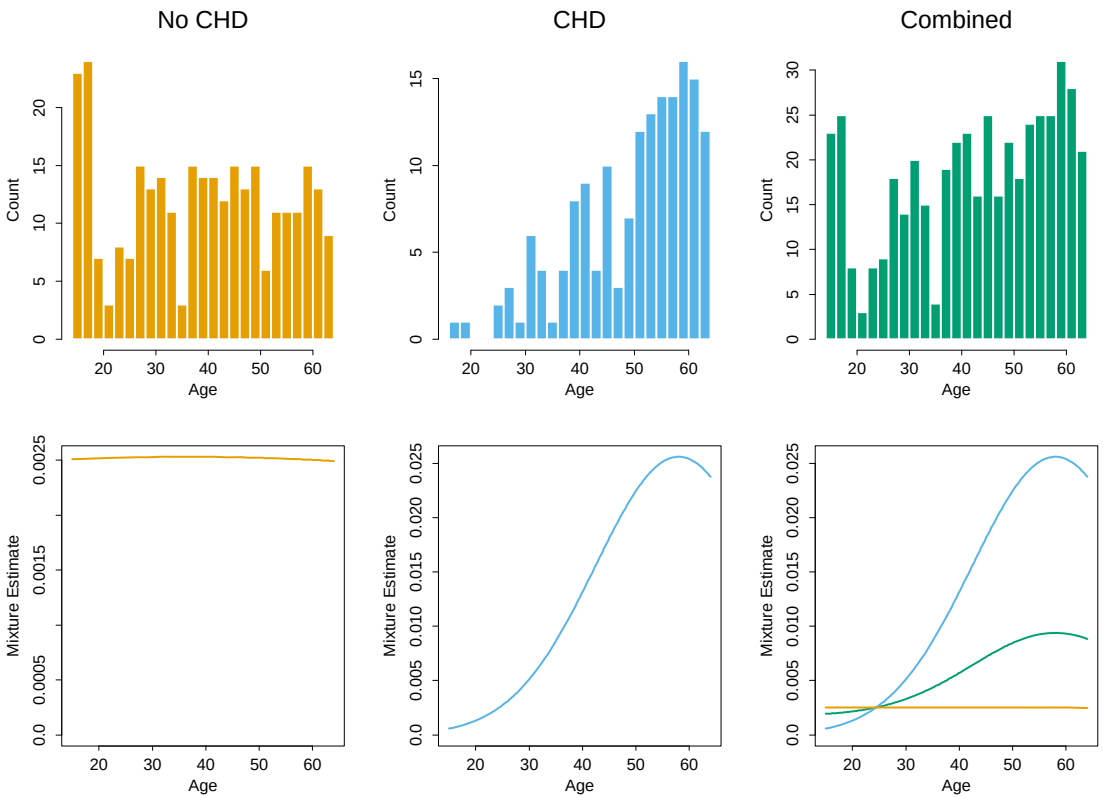
$$f(x) = \sum_{j=1}^{M} K_{\lambda_j}(\xi_j, x)\beta_j,$$

- We should either fit the lengthscale parameters λ_j or re-normalize the radial basis functions. Otherwise, the RBF can leave holes (upper figure, re-normalized down).



Mixuture Models for Density Estimation and Classification

- One RBF kernel was fitted for each class
- The data sample is classified according the more probable label (let kernels vote).
- If the covariance matrices are constrained to be scalar $\Sigma_m = \sigma_m I$, we actually fit the naive Bayes model.
- In this case, this method was as good as logistic regression.



Summary

- K-means clustering - the basic one
  - the number of clusters:
  - GAP
  - Silhouette
- The distance is crucial.
  - Consider standardization or weighting the features.
- K-medoids - does need metric, just a distance
- hierarchical clustering
  - different distance measures
  - dendrogram
- other approaches (mixture of Gaussians, Dirichlet Process Mixture Model, mean shift clustering, Self Organizing Maps, Spectral Clustering).

Mean Shift Clustering (from here just notes)

Mean Shift Clustering

1: **procedure** MEAN SHIFT CLUSTERING:(X data, $K(\cdot)$ the kernel, λ the bandwidth)
2: $\mathcal{C} \leftarrow \emptyset$
3: **for** each data record **do**
4: **repeat** # shift each mean x to the weighted average
5: $m(x) \leftarrow \frac{\sum_{i=1}^{N} K(x_i - x) x_i}{\sum_{i=1}^{N} K(x_i - x)}$
6: **until** no chance in assignment
7: add the new $m(x)$ to $\mathcal{C}$
8: **end for**
9: return prunned $\mathcal{C}$
10: **end procedure**

Kernels:

- flat kernel λ ball
- Gaussian kernel $K(x_i - x) = e^{\frac{\|x_i - x\|^2}{\lambda^2}}$

List of topics

1. Linear, ridge, lasso regression, k-neares neighbours,(formulas) overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, generalized additive models
4. Train/test error and data split, square error, 0-1, crossentropy, AIC, BIC,(formulas) crossvalidation, one-leave-out CV, wrong estimate example
5. decision trees, information gain/entropy/gini, CART prunning,(formulas)
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost(formulas) and gradient boosting), stacking, MARS,
7. Bayesian learning: MAP, ML hypothesis (formulas), Bayesian optimal prediction, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, deviance, MRF
12. Gaussian processes: estimation of the function and its variance (figures, ideas).

List of topics

1. Linear, ridge, lasso regression, k-neares neighbours,(formulas) overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, generalized additive models
4. Train/test error and data split, square error, 0-1, crossentropy, AIC, BIC,(formulas) crossvalidation, one-leave-out CV, wrong estimate example
5. decision trees, information gain/entropy/gini, CART prunning,(formulas)
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost(formulas) and gradient boosting), stacking, MARS,
7. Bayesian learning: MAP, ML hypothesis (formulas), Bayesian optimal prediction, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, deviance, MRF
12. Gaussian processes: estimation of the function and its variance (figures, ideas).

Table of Contens