Recitation – Soft k-means clustering

Hongyu & Wendy

Review of lecture material

Exploring population structure

Phylogenetic trees

Classification problem
(k-nearest neighbor)

Clustering
(k-means clustering)

May apply PCA first

K-means clustering – Lloyd Algorithm

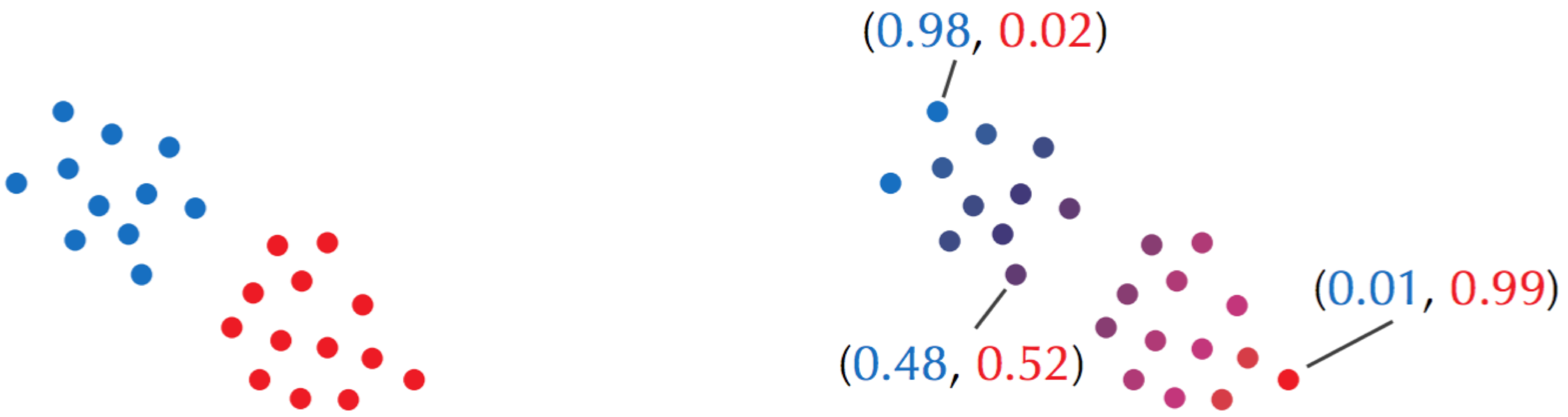
Select *k* arbitrary data points as *Centers* and then iteratively perform the following steps:

- **Centers to Clusters**: Assign each data point to the cluster corresponding to its nearest center (ties are broken arbitrarily).
- **Clusters to Centers**: After the assignment of data points to *k* clusters, compute new centers as clusters' center of gravity.

K-means clustering– Lloyd Algorithm

Observation: Centers and clusters are both hidden and we try to infer them in stages ... just like EM/Gibbs!

Admixture - From hard to soft



Hard choices: points are colored red or blue depending on their cluster membership.

Soft choices: points are assigned "red" and "blue" *responsibilities* r_{blue} and r_{red} ($r_{blue} + r_{red} = 1$)

From hard to soft

Select *k* arbitrary data points as *Centers* and then iteratively perform the following steps:

- **Centers to Clusters**: Assign each data point to the ~~cluster corresponding to its nearest center (ties are broken arbitrarily~~).
  a 'responsibility' value for each cluster
- **Clusters to Centers**: After the assignment of data points to *k* clusters, compute new centers as clusters' center of gravity.

Soft k-means clustering

- **Centers to Soft Clusters (E-step):** After centers have been selected, assign each data point a "responsibility" value for each cluster, where higher values correspond to stronger cluster membership.
- **Soft Clusters to Centers (M-step):** After data points have been assigned to soft clusters, compute new centers.

Centers to soft clusters

Calculate HiddenMatrix

Input: Given k centers *Centers* = (x_1, ..., x_k) and n points *Data* = ($Data_1$, ... , $Data_n$)

Output: Construct a $k \times n$ responsibility matrix *HiddenMatrix* for which $HiddenMatrix_{i,j}$ is the pull of center i on data point j.

Centers to soft clusters

Think about centers as stars and data points as planets

By Newtonian inverse-square law of gravitation:

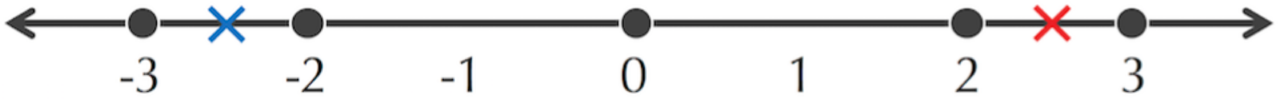
$$HiddenMatrix_{i,j} = \frac{1/d(Data_j, x_i)^2}{\sum_{\text{all centers } x_t} 1/d(Data_j, x_t)^2}.$$

In practice this works better:

$$HiddenMatrix_{i,j} = \frac{e^{-\beta \cdot d(Data_j,\, x_i)}}{\sum_{\text{all centers } x_t} e^{-\beta \cdot d(Data_j,\, x_t)}}.$$

β is a parameter reflecting the amount of flexibility in our soft assignment and called the **stiffness parameter**.

Centers to soft clusters



0.992	0.988	0.500	0.012	0.008	Newtonian
0.008	0.012	0.500	0.988	0.992	
0.500	0.500	0.500	0.500	0.500	$\beta = 0$
0.500	0.500	0.500	0.500	0.500	
0.924	0.881	0.500	0.119	0.076	$\beta = 0.5$
0.076	0.119	0.500	0.881	0.924	
0.993	0.982	0.500	0.018	0.007	$\beta = 1$
0.007	0.018	0.500	0.982	0.993	
1.000	1.000	0.500	0.000	0.000	$\beta = 1000$
0.000	0.000	0.500	1.000	1.000	

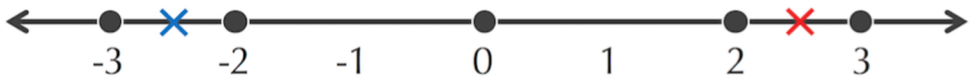
Soft clusters to centers

M-step: Update weighed center of gravity

$x_{i,\,j}$ -- j-th coordinate of center x_i

$$x_{i,j} = \frac{HiddenMatrix_i \cdot Data^j}{HiddenMatrix_i \cdot \overrightarrow{1}}$$

Soft clusters to centers



0.992	0.988	0.500	0.012	0.008	Newtonian
0.008	0.012	0.500	0.988	0.992	
0.500	0.500	0.500	0.500	0.500	β = 0
0.500	0.500	0.500	0.500	0.500	
0.924	0.881	0.500	0.119	0.076	β = 0.5
0.076	0.119	0.500	0.881	0.924	
0.993	0.982	0.500	0.018	0.007	β = 1
0.007	0.018	0.500	0.982	0.993	

$$x_1 = \frac{0.993 \cdot (-3) + 0.982 \cdot (-2) + 0.500 \cdot (0) + 0.018 \cdot (2) + 0.007 \cdot (3)}{0.993 + 0.982 + 0.500 + 0.018 + 0.007} = -1.955$$

$$x_2 = \frac{0.007 \cdot (-3) + 0.018 \cdot (-2) + 0.500 \cdot (0) + 0.982 \cdot (2) + 0.993 \cdot (3)}{0.007 + 0.018 + 0.500 + 0.982 + 0.993} = 1.955$$