BIOSTAT 527 4/19/23

Lecture 8

- Future topics
- K-Means, EM clustering

More clustering?

1) Cost-based (K-means)
2) Mixtures (of Gaussians)

} parametric

Hierarchical clustering with costs 1) and 2)

Vote:

Name yes/No

YES if $\geq 50\% + 1$ yes in class

expensive n^2

Kernel Machines → SVM
→ GP (Gauss Processes)

Kernel ⇒ Cov Σ

like KDE

Double Descent Benefic Overfitting

NN as GP, NTK

Random Fourier ~ n

RKHS

Combining randomized predictors

Decision Trees ←

Random Forests independently

Boosting ← sequentially

* Bootstrap 1 lec

NP - Bayes
+ DProcess → Clustering

Point process:
- Determinantal PP

KDE → Manifold Learning < concepts, algorithms

Non-linear dim reduction

Lecture VIII: Classic and Modern Data Clustering – Part I

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

May, 2022

Paradigms for clustering

Parametric clustering algorithms (K given)

Cost based / hard clustering

k-means

Basic algorithms

K-means clustering and the quadratic distortion

Model based / soft clustering

Mixtures

Issues in parametric clustering

Selecting K

Reading: 14.3Ch 11.[1], 11.2.1-3, 11.3, Ch 25

What is clustering? Problem and Notation

- **Informal definition** **Clustering** = Finding groups in data
- **Notation**

$$\begin{aligned} \mathcal{D} &= \{x_1, x_2, \ldots x_n\} \text{ a } \textbf{data set} \\ n &= \text{number of } \textbf{data points} \\ K &= \text{number of } \textbf{clusters } (K << n) \\ \Delta &= \{C_1, C_2, \ldots, C_K\} \text{ a partition of } \mathcal{D} \text{ into disjoint subsets} \\ k(i) &= \text{the } \textbf{label} \text{ of point } i \\ \mathcal{L}(\Delta) &= \text{cost (loss) of } \Delta \text{ (to be minimized)} \end{aligned}$$

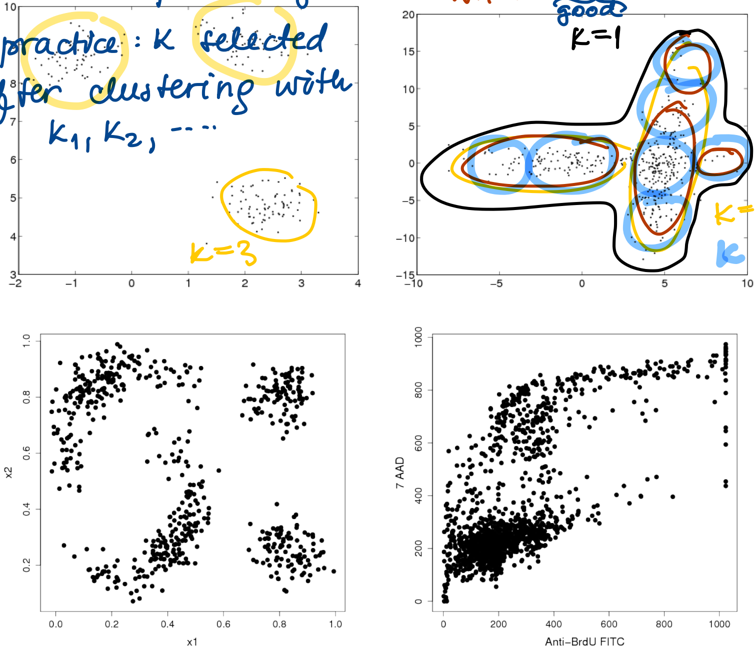
- **Second informal definition** **Clustering** = given n **data points**, separate them into K **clusters**
- Hard vs. soft clusterings
  - **Hard** clustering Δ: an item belongs to only 1 cluster
  - **Soft** clustering $\gamma = \{\gamma_{ki}\}_{k=1:K}^{i=1:n}$
    γ_{ki} = the **degree of membership** of point i to cluster k

$$\sum_k \gamma_{ki} = 1 \quad \text{for all } i$$

(usually associated with a probabilistic model)

Paradigm = what is good cluster(ing)?
what algorithm outputs good clusterings?

- K given → input to alg
- in practice: K selected after clustering with $K_1, K_2, \ldots$
- What is a good cluster?



(from [?])

Paradigms

Depend on type of data, type of clustering, type of cost (probabilistic or not), and constraints (about K, shape of clusters)

- Data = vectors $\{x_i\}$ in $\mathbb{R}^d$

Parametric	Cost based [hard]
(K known)	Model based [soft]
Non-parametric	Dirichlet process mixtures [soft]
(K determined	Information bottleneck [soft]
by algorithm)	Modes of distribution [hard]
	Gaussian blurring mean shift[?] [hard]

implicit: what is good cluster

probabilistic model for each cluster e.g Gaussian

- Data = similarities between pairs of points $[S_{ij}]_{i,j=1:n}$, $S_{ij} = S_{ji} \geq 0$

Similarity based clustering

Graph partitioning	spectral clustering [hard, K fixed, cost based]
	typical cuts [hard non-parametric, cost based]
Affinity propagation	[hard/soft non-parametric]

Classification vs Clustering

	Classification	**Clustering**
Cost (or Loss) $\mathcal{L}$	Expectd error	many! (probabilistic or not)
	Supervised	Unsupervised
Generalization	Performance on new data is what matters	Performance on current data is what matters
K	Known	Unknown
"Goal"	Prediction	Exploration Lots of data to explore!
Stage of field	Mature	Still young

Parametric clustering algorithms

- Cost based
  - Single linkage (min spanning tree)
  - Min diameter
    - Fastest first traversal (HS initialization)
  - K-medians
  - K-means Least Squares
- Model based (cost is derived from likelihood)
  - EM algorithm Mixtures
  - "Computer science"/"Probably correct" algorithms

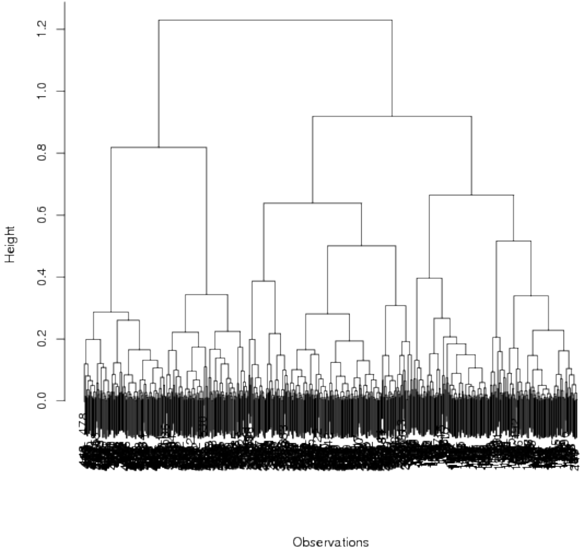
Single Linkage Clustering

Algorithm Single-Linkage

Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, number clusters K

1. Construct the Minimum Spanning Tree (MST) of $\mathcal{D}$
2. Delete the largest $K-1$ edges

- **Cost** $\mathcal{L}(\Delta) \; = \; -\min_{k,k'} \; \mathrm{distance}(C_k, C_{k'})$
  where $\mathrm{distance}(A, B) = \underset{x \in A,\, y \in B}{\mathrm{argmin}} \, ||x - y||$
- Running time $\mathcal{O}(n^2)$ one of the very few costs $\mathcal{L}$ that can be optimized in **polynomial** time
- Sensitive to outliers!


Height
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Observations

Minimum diameter clustering

- **Cost** $\mathcal{L}(\Delta) = \max_k \underbrace{\max_{i,j \in C_k} ||x_i - x_j||}_{\text{diameter}}$
  - Mimimize the diameter of the clusters
  - Optimizing this cost is NP-hard
- **Algorithms**
  - **Fastest First Traversal** [?] – a factor 2 approximation for the min cost

For every $\mathcal{D}$, FFT produces a Δ so that

$$\mathcal{L}^{opt} \leq \mathcal{L}(\Delta) \leq 2\mathcal{L}^{opt}$$

  - rediscovered many times

Algorithm Fastest First Traversal

Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, number clusters K

defines **centers** $\mu_{1:K} \in \mathcal{D}$

(many other clustering algorithms use centers)

1. pick μ_1 at random from $\mathcal{D}$
2. for $k = 2 : K$

$$\mu_k \leftarrow \underset{\mathcal{D}}{\operatorname{argmax}} \operatorname{distance}(x_i, \{\mu_{1:k-1}\})$$

3. for $i = 1 : n$ (assign points to centers)

$k(i) = k$ if μ_k is the nearest center to x_i

K-means clustering

$\mu_{1,2,\cdots k}$ = representatives for $C_{1,2,k}$

Algorithm K-Means[?]

Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, number clusters K
tialize **centers** $\mu_1, \mu_2, \ldots \mu_K \in \mathbb{R}^d$ at random
terate until convergence

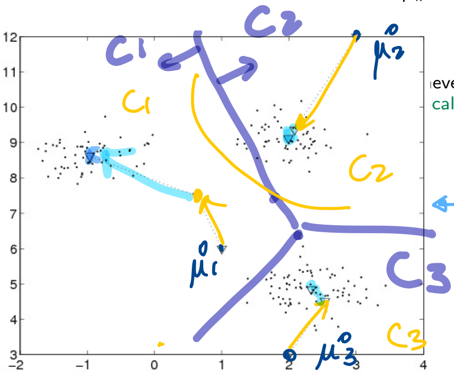
1. for $i = 1 : n$ (assign points to clusters $\Rightarrow$ new clustering)

$$\underbrace{k(i)}_{\text{label of } x^i} = \operatorname*{argmin}_k ||x_i - \mu_k||$$

2. for $k = 1 : K$ (recalculate centers)

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i \qquad (1)$$

recalculate μ_k



ever change after that
cal optimum of cost $\mathcal{L}$ (defined next)

Converged in 3 steps

The K-means cost → "loss"

sum squared distances to μ_k

$$\text{cost} \quad \mathcal{L}(\Delta) = \sum_{k=1}^{K} \boxed{\sum_{i \in C_k} ||x_i - \mu_k||^2} \tag{2}$$

- K-means solves a **least-squares** problem
- the cost $\mathcal{L}$ is called **quadratic distortion**

1. **Proposition** The K-means algorithm decreases $\mathcal{L}(\Delta)$ at every step.
2. Converges in finite steps
3. .. to a local optimum of $\mathcal{L}(\Delta)$
4. Corollary: initialization matters !!

smart initialization
multiple runs

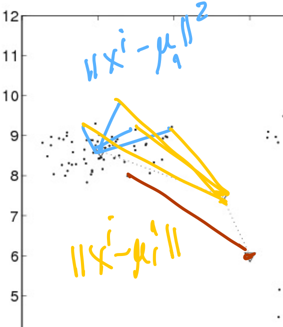
Sketch of proof

- step 1: reassigning the labels can only decrease $\mathcal{L}$
- step 2: reassigning the centers μ_k can only decrease $\mathcal{L}$ because μ_k as given by (1) is the solution to

Exercise

$$\mu_k = \min_{\mu \in \mathbb{R}^d} \sum_{i \in C_k} ||x_i - \mu||^2$$

$$\mu_k = \operatorname*{mean}_{x^i \in C_k} (x^i)$$



Equivalent and similar cost functions

- The distortion can also be expressed using intracluster distances

$$\mathcal{L}(\Delta) = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,j \in C_k} ||x_i - x_j||^2 \tag{4}$$

- **Correlation clustering** is defined as optimizing the related criterion

$$\mathcal{L}(\Delta) = \sum_{k=1}^{K} \sum_{i,j \in C_k} ||x_i - x_j||^2$$

- This cost is equivalent to the (negative) sum of (squared) intercluster distances

$$\mathcal{L}(\Delta) = -\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \notin C_k} ||x_i - x_j||^2 + \text{constant} \tag{5}$$

Proof of (6) Replace μ_k as expressed in (1) in the expression of $\mathcal{L}$, then rearrange the terms

Proof of (5) $\sum_k \sum_{i,j \in C_k} ||x_i - x_j||^2 = \underbrace{\sum_{i=1}^{n} \sum_{j=1}^{n} ||x_i - x_j||^2}_{\text{independent of } \Delta} - \sum_k \sum_{i \in C_k} \sum_{j \notin C_k} ||x_i - x_j||^2$

The K-means cost in matrix form – the assignment matrix

- $\mathcal{L}$ as sum of squared intracluster distances

$$\mathcal{L}(\Delta) \;=\; \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,j \in C_k} ||x_i - x_j||^2 \tag{6}$$

-
- Define the **assignment matrix** associated with Δ by $Z(\Delta)$
  Let $\Delta = \{C_1 = \{1,2,3\},\; C_2 = \{4,5\}\}$

$$Z^{unnorm}(\Delta) \;=\; \begin{array}{cc} C_1 & C_2 \\ \end{array} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \text{point } i \qquad Z(\Delta) \;=\; \begin{array}{cc} C_1 & C_2 \\ \end{array} \begin{bmatrix} 1/\sqrt{3} & 0 \\ 1/\sqrt{3} & 0 \\ 1/\sqrt{3} & 0 \\ 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{bmatrix}$$

Then Z is an orthogonal matrix (columns are orthornormal) and

$$\mathcal{L}(\Delta) \;=\; \operatorname{trace} Z^T D Z \qquad \text{with } D_{ij} = ||x_i - x_j||^2 \tag{7}$$

$$\text{Let } \mathcal{Z} \;=\; \{\, Z \in \mathbb{R}^{n \times K},\; K \text{ orthonormal} \,\}$$

Proof of (7) Start from (2) and note that $\operatorname{trace} Z^T A Z = \sum_k \sum_{i,j \in C_k} Z_{ik} Z_{jk} A_{ij} = \sum_k \sum_{i,j \in C_k} \frac{1}{|C_k|} A_{ij}$

The K-means cost in matrix form – the co-ocurrence matrix

$$n = 5,\ \Delta = (1, 1, 1, 2, 2), \qquad X(\Delta) = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

1. $X(\Delta)$ is symmetric, positive definite, ≥ 0 elements
2. $X(\Delta)$ has row sums equal to 1
3. $\operatorname{trace} X(\Delta) = K$

$$\begin{aligned} \|X(\Delta)\|_F^2 &= \langle X, X \rangle = K \\ X(\Delta) &= Z(\Delta) Z^T(\Delta) \end{aligned}$$

$$2\mathcal{L}(\Delta) = \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,j \in C_k} ||x_i - x_j||^2 = \frac{1}{2} \langle D, X(\Delta) \rangle$$

with $D_{ij} = ||x_i - x_j||^2$

Spectral and convex relaxations

$$
\begin{aligned}
\mathcal{L}(\Delta) &= \frac{1}{2}\langle D, X(\Delta)\rangle, \quad D = \text{squared distance matrix} \in \mathbb{R}^{n\times n} \\
\mathcal{X} &= \{\, X \in \mathbb{R}^{n\times n},\ X \succeq 0,\ X_{ij} \geq 0,\ \operatorname{trace} X = K,\ X1 = 1 \,\} \\
\mathcal{Z} &= \{\, Z \in \mathbb{R}^{n\times K},\ K \text{ orthonormal} \,\}
\end{aligned}
$$

Spectral relaxation of the K-means problem

$$\min_{Z\in\mathcal{Z}} \operatorname{trace} Z^T DZ$$

This is solved by an **eigendecomposition** $Z^* =$ top K eigenvectors of D

Convex relaxation of the K-means problem

$$\min_{X\in\mathcal{X}} \langle D, X\rangle$$

This is a **Semi-Definite Program (SDP)**
Minimizing $\mathcal{L}$

- By K-means – clustering Δ, local optima
- By convex/spectral relaxation – matrix Z, X, global optimum

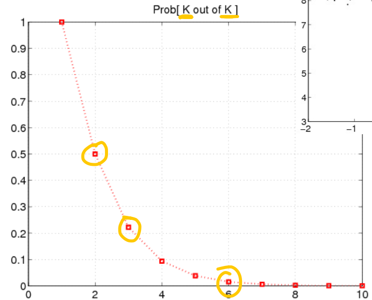
Symmetries between costs

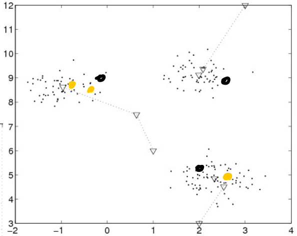
- K-means cost $\mathcal{L}(\Delta) \;=\; \min_{\mu_{1:K}} \sum_k \sum_{i \in C_k} ||x_i - \mu_k||^2$
- K-medians cost $\mathcal{L}(\Delta) \;=\; \min_{\mu_{1:K}} \sum_k \sum_{i \in C_k} ||x_i - \mu_k||$

- Correlation clustering cost $\mathcal{L}(\Delta) \;=\; \sum_k \sum_{i,j \in C_k} ||x_i - x_j||^2$
- min Diameter cost $\mathcal{L}^2(\Delta) \;=\; \max_k \max_{i,j \in C_k} ||x_i - x_j||^2$

Initialization of the centroids $\mu_{1:K}$

- Idea 1: start with K points at random
- Idea 2: start with K data points at random
  What's wrong with chosing K data points at random?





The probability of hitting all K clusters with K samples approaches 0 when $K > 5$

- Idea 3: start with K data points using Fastest First Traversal [] (greedy simple approach to spread out centers)
- Idea 4: **k-means++** [] (randomized, theoretically backed approach to spread out centers)
- Idea 5: **"K-logK" Initialization** (start with enough centers to hit all clusters, then prune down to K)
  For EM Algorithm [], for K-means [**?**]

The “K-logK” initialization

The K-logK Initialization (see also [**?**])

1. pick $\mu^0_{1:K'}$ at random from data set, where $K' = O(K \log K)$
   (this assures that each cluster has at least 1 center w.h.p)
2. run 1 step of K-means
3. remove all centers μ^0_k that have few points, e.g $|C_k| < \frac{n}{eK'}$
4. from the remaining centers select K centers by **Fastest First Traversal**
   4.1 pick μ_1 at random from the remaining $\{\mu^0_{1:K'}\}$
   4.2 for $k = 2 : K$, $\mu_k \leftarrow \underset{\mu^0_{k'}}{\text{argmax}} \min_{j=1:k-1} ||\mu^0_{k'} - \mu_j||$, i.e next μ_k is furthest away from the already chosen centers
5. continue with the standard **K-means** algorithm

The "kmeans++" initialization

1. pick μ_1 uniformly at random from the data
2. for $k = 2 : K$,
   - Define a distribution over data $x_{1:n}$ by

$$P_k(x_i) \propto \min_{j=1:k-1} ||x_i - \mu_j||^2$$

   - Sample $\mu_k \sim P_k$ (i.e next μ_k is probabilistically far away from the already chosen centers

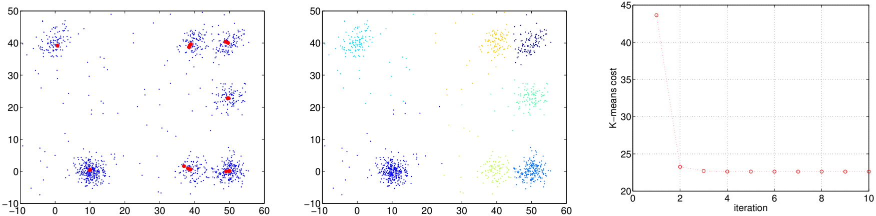
Comparison between FFT, K-logK, kmeans++

- all three methods can be seen as variants of FFT
- FFT alone tends to choose outliers
- K-logK and kmeans++ can be seen as robust forms of FFT
- K-logK guarantees w.h.p. that no outliers will be chosen (by elimnating all small clusters)
- the most expensive step in K-logK method is the first K-means step, which takes $nK \log(K)$ distance computations
- the computational cost of kmeans++ is $(K-1)n$ distance computations and $Kn \log(n)$ for sampling from $P_{2:K}$

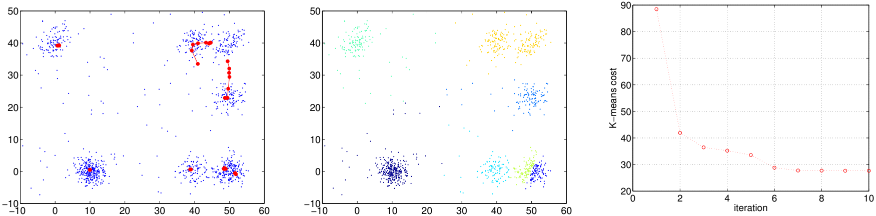
K-means clustering with K-logK Initialization

Example using a mixture of 7 Normal distributions with 100 outliers sampled uniformly

K-LOGK $K = 7,\ T = 100,\ n = 1100,\ c = 1$



NAIVE $K = 7\ \ T = 100,\ n = 1100$



Coresets approach to K-medians and K-means

- A weighted subset of $\mathcal{D}$ is a (K, ε) **coreset** iff for any $\mu_{1:K}$,

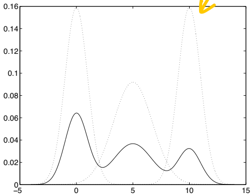
$$|\mathcal{L}(\mu_{1:K}, A) - \mathcal{L}(\mu_{1:K}; \mathcal{D})| \leq \varepsilon \mathcal{L}(\mu_{1:K}; \mathcal{D})$$

- Note that the size of A is not K
- Finding a coreset (fast) lets use find fast algorithms for clustering a large $\mathcal{D}$
  - "fast" $=$ linear in n, exponential in ε^{-d}, polynomial in K
- **Theorem**[?],Theorem 5.7
  One can compute an $(1+\varepsilon)$-approximate K-median of a set of n points in time $\mathcal{O}(n + K^5 \log^9 n + gK^2 \log^5 n)$ where $g = e^{[C/\varepsilon \log(1+1/\varepsilon)]^{d-1}}$ (where d is the dimension of the data)
- **Theorem**[?],Theorem 6.5
  One can compute an $(1+\varepsilon)$-approximate K-means of a set of n points in time $\mathcal{O}(n + K^5 \log^9 n + K^{K+2}\varepsilon^{-(2d+1)} \log^{K+1} n \log^K \frac{1}{\varepsilon})$.

Model based clustering: Mixture models

= density estimation

Mixture in 1D



components

- The **mixture density**

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x)$$

a density

representative of C

- $f_k(x)$ = the **components** of the mixture
  - each is a density
  - f called **mixture of Gaussians** if $f_k = Normal_{\mu_k, \Sigma_k}$
- π_k = the **mixing proportions**, $\sum_k = 1^K \pi_k = 1, \quad \pi_k > 0$.
- **model parameters** $\theta = (\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K})$
- The **degree of membership** of point i to cluster k

$$\gamma_{ki} \stackrel{\text{def}}{=} P[x_i \in C_k] = \frac{\pi_k f_k(x)}{f(x)} \text{ for } i = 1 : n, k = 1 : K \tag{8}$$

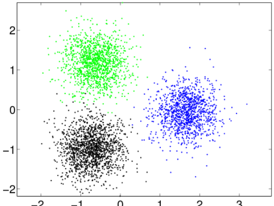
- depends on x_i and on the model parameters

$\pi : \{1, .. K\}$ $\quad \pi_k = Pr[k]$

Gaussians $f_{1,..K}$

1. Sample $k \sim \pi$. 2. Sample $x \sim f_k$

Mixture in 2D



Degree of membership

$$\gamma_k(i) = \Pr[\, x^i \text{ was sampled from } f_k \,]$$

Bayes $\longrightarrow$ $$= \frac{\pi_k f_k(x^i)}{\sum_{k'} \pi_{k'} f_{k'}(x^i)}$$