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

Basic algorithms

K-means clustering and the quadratic distortion

Model based / soft clustering

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

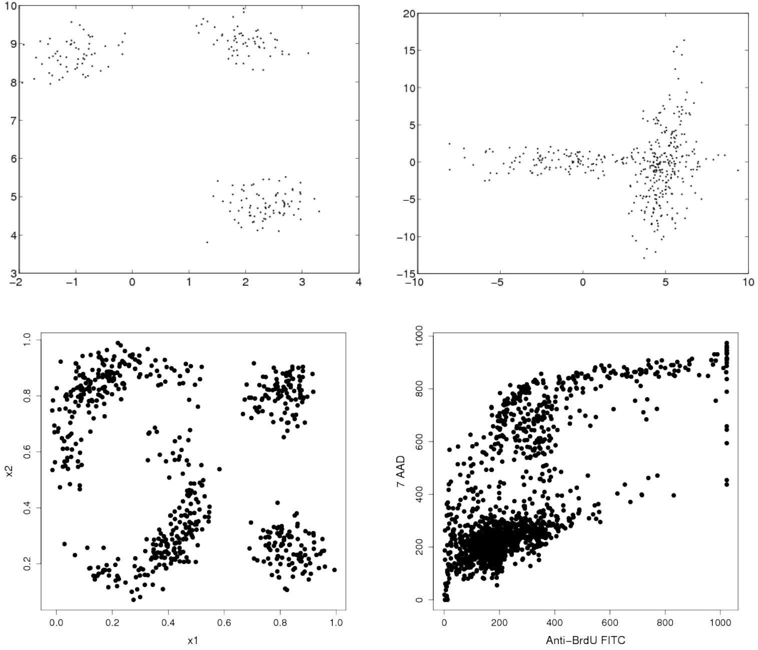
- **Informal definition** **Clustering** = Finding groups in data
- **Notation**

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

- **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)



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

- 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
- Model based (cost is derived from likelihood)
  - EM algorithm
  - "Computer science"/"Probably correct" algorithms

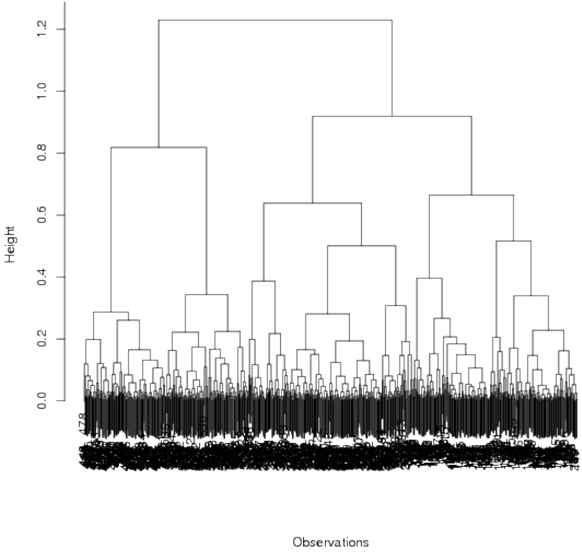
Single Linkage Clustering

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** $\mathcal{L}(\Delta) = -\min_{k,k'} \text{distance}(C_k, C_{k'})$
  where $\text{distance}(A, B) = \underset{x \in A,\, y \in B}{\text{argmin}} ||x - y||$
- 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-medians clustering

- **Cost** $\mathcal{L}(\Delta) = \sum_k \sum i \in C_k ||x_i - \mu_k||$ with $\mu_k \in \mathcal{D}$
  - (usually) assumes centers chosen from the data points (analogy to median)

Exercise Show that in 1D $\operatorname{argmin}_{\mu} \sum_i |x_i - \mu|$ is the median of $\{x_i\}$

  - optimizing this cost is NP-hard
  - has attracted a lot of interest in theoretical CS (general from called "Facility location"

Integer Programming Formulation of K-medians

- Define $d_{ij} = ||x_i - x_j||$,
  $u_{ij} = 1$ iff point i in cluster with center x_j (0 otherwise),
  $y_j = 1$ iff point j is cluster center (0 otherwise)

$$
\begin{array}{lllll}
\min\limits_{u,y} & \sum_{ij} d_{ij} u_{ij} & & & \\
\text{s.t.} & \sum_j u_{ij} & = & 1 & \text{point } i \text{ is in exactly 1 cluster for all } i \\
 & \sum_j y_j & \leq & k & \text{there are at most } k \text{ clusters} \\
 & u_{ij} & \leq & y_j & \text{point } i \text{ can only belong to a center for all } i,j
\end{array}
$$

Linear Programming Relaxation of K-medians

- Define d_{ij}, $y_j = 1, u_{ij}$ as before, but $y_j, u_{ij} \in [0, 1]$

$$
\textbf{(LP)} \quad
\begin{array}{llll}
\min\limits_{u,y} & \sum_{ij} d_{ij} u_{ij} & & \\
\text{s.t.} & \sum_j u_{ij} & = & 1 \\
 & \sum_j y_j & \leq & k \\
 & u_{ij} & \leq & y_j
\end{array}
$$

Algorithm K-Medians (variant of [?])

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

1. Solve (LP)

   obtain fractionary "centers" $y_{1:n}$ and "assignments" $u_{1:n,1:n}$

2. Sample K centers $\mu_1 \ldots \mu_K$ by
   - $P[\mu_k = \text{point}j] \propto y_j$ (without replacement)

3. Assign points to centers (deterministically)

$$k(i) \; = \; \operatorname*{argmin}_k ||x_i - \mu_k||$$

- Guarantees (Agarwal)
  - **Given** tolerance ε, confidence δ, $K' = K(1 + \frac{1}{\varepsilon}) \ln \frac{n}{K}$, $\Delta_{K'}$ obtained by **K-medians** with K' centers

$$\mathcal{L}(\Delta_{K'}) \; \leq \; (1 + \varepsilon)\mathcal{L}_K^{opt}$$

K-means clustering

Algorithm K-Means[?]

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

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

$$k(i) = \underset{k}{\operatorname{argmin}} ||x_i - \mu_k||$$

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

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i \tag{1}$$

- Convergence
  - if Δ doesn't change at iteration m it will never change after that
  - convergence in finite number of steps to local optimum of cost $\mathcal{L}$ (defined next)
  - therefore, initialization will matter

The K-means cost

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

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

Proposition The K-means algorithm decreases $\mathcal{L}(\Delta)$ at every step.

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

$$\mu_k = \min_{\mu \in \mathbb{R}^d} \sum_{i \in C_k} ||x_i - \mu||^2 \tag{3}$$

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

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

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

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

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

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

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

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

$n = 5,\ \Delta = (1, 1, 1, 2, 2),$

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

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

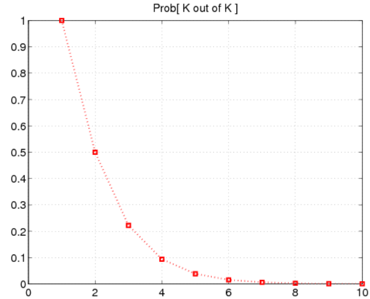
Symmetries between costs

- 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 $k = 2 : K$, $\mu_k \leftarrow \operatorname{argmax}_{\mu^0_{k'}} \min_{j=1:k-1} ||\mu^0_{k'} - \mu_j||$, 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

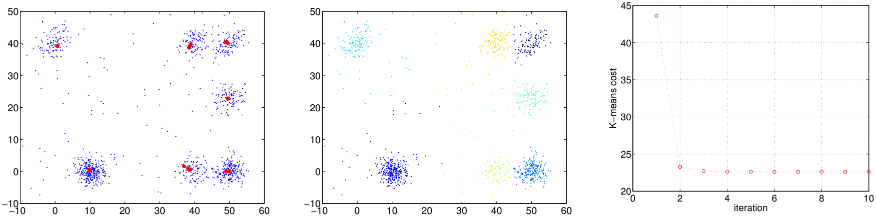
Comparison between FFT, K-logK, kmeans++

- 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 $P_{2:K}$

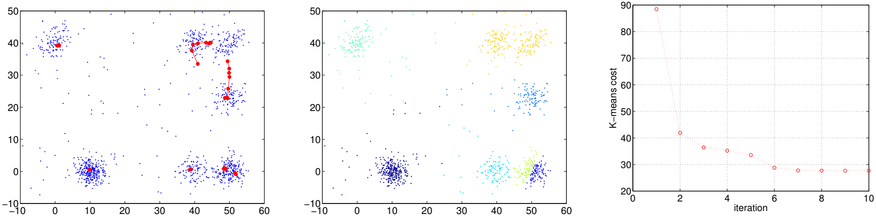
K-means clustering with K-logK Initialization

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

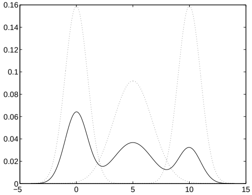
- A weighted subset of $\mathcal{D}$ is a (K, ε) **coreset** iff for any $\mu_{1:K}$,

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

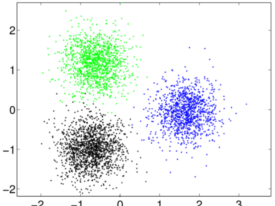
- Note that the size of A is not K
- Finding a coreset (fast) lets use find fast algorithms for clustering a large $\mathcal{D}$
  - "fast" $=$ linear in n, exponential in ε^{-d}, polynomial in K
- **Theorem**[?],Theorem 5.7
  One can compute an $(1+\varepsilon)$-approximate K-median of a set of n points 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

Mixture in 1D



Mixture in 2D



- The **mixture density**

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

- $f_k(x)$ = 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**, $\sum_k = 1^K \pi_k = 1, \;\; \pi_k \geq 0$.
- **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

Criterion for clustering: Max likelihood

- denote $\theta = (\pi_{1:K},\, \mu_{1:K},\, \Sigma_{1:K})$ (the parameters of the mixture model)
- Define **likelihood** $P[\mathcal{D}|\theta] \;=\; \prod_{i=1}^{n} f(x_i)$
- Typically, we use the **log likelihood**

$$l(\theta) \;=\; \ln \prod_{i=1}^{n} f(x_i) \;=\; \sum_{i=1}^{n} \ln \sum_{k} \pi_k f_k(x_i) \tag{9}$$

- denote $\theta^{ML} \;=\; \operatorname*{argmax}_{\theta} l(\theta)$
- θ^{ML} determines a soft clustering γ by (8)
- a soft clustering γ determines a θ (see later)
- Therefore we can write

$$\mathcal{L}(\gamma) \;=\; -l(\theta(\gamma))$$

Algorithms for model-based clustering

Maximize the (log-)likelihood w.r.t θ

- directly - (e.g by gradient ascent in θ)
- by the EM algorithm (very popular!)
- indirectly, w.h.p. by "computer science" algorithms

w.h.p = with high probability (over data sets)

The Expectation-Maximization (EM) Algorithm

Algorithm Expectation-Maximization (EM)

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

tialize parameters $\pi_{1:K} \in \mathbb{R}$, $\mu_{1:K} \in \mathbb{R}^d$, $\Sigma_{1:K} \in \mathbb{R}^{d \times d}$ at random[1]

terate until convergence

E step (Optimize clustering) for $i = 1 : n$, $k = 1 : K$

$$\gamma_{ki} = \frac{\pi_k f_k(x)}{f(x)}$$

M step (Optimize parameters) set $\Gamma_k = \sum_{i=1}^n \gamma_{ki}$, $k = 1 : K$ (number of points in cluster k)

$$\pi_k = \frac{\Gamma_k}{n}, \quad k = 1 : K$$

$$\mu_k = \sum_{i=1}^n \frac{\gamma_{ki}}{\Gamma_k} x_i$$

$$\Sigma_k = \frac{\sum_{i=1}^n \gamma_{ki}(x_i - \mu_k)(x_i - \mu_k)^T}{\Gamma_k}$$

- $\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K}$ are the maximizers of $l_c(\theta)$ in (13)
- $\sum_k \Gamma_k = n$

[1] Σ_k need to be symmetric, positive definite matrices

The EM Algorithm – Motivation

- Define the **indicator variables**

$$z_{ik} = \begin{cases} 1 & \text{if } i \in C_k \\ 0 & \text{if } i \notin C_k \end{cases} \tag{10}$$

  denote $\bar{z} = \{z_{ki}\}_{k=1:K}^{i=1:n}$
- Define the **complete log-likelihood**

$$l_c(\theta, \bar{z}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ki} \ln \pi_k f_k(x_i) \tag{11}$$

- $E[z_{ki}] = \gamma_{ki}$
- Then

$$E[l_c(\theta, \bar{z})] = \sum_{i=1}^{n} \sum_{k=1}^{K} E[z_{ki}][\ln \pi_k + \ln f_k(x_i)] \tag{12}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ki} \ln \pi_k + \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ki} \ln f_k(x_i)] \tag{13}$$

- If θ known, γ_{ki} can be obtained by (8) **(Expectation)**
- If γ_{ki} known, π_k, μ_k, Σ_k can be obtained by separately maximizing the terms of $E[l_c]$ **(Maximization)**

Brief analysis of EM

$$Q(\theta, \gamma) \;=\; \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ki} \ln \underbrace{\pi_k f_k(x_i)}_{\theta}$$

- each step of EM increases $Q(\theta, \gamma)$
- Q converges to a local maximum
- at every local maxi of Q, $\theta \leftrightarrow \gamma$ are fixed point
- $Q(\theta^*, \gamma^*)$ local max for $Q \Rightarrow\; l(\theta^*)$ local max for $l(\theta)$
- under certain regularity conditions $\theta \longrightarrow \theta^{ML}$ [**?**]
- the E and M steps can be seen as projections [**?**]

- Exact maximization in **M step** is not essential.
  Sufficient to increase Q.
  This is called **Generalized EM**

Probablistic alternate projection view of EM[?]

- let z_i = which gaussian generated i? (random variable), $X = (x_{1:n})$, $Z = (z_{1:n})$
- Redefine Q

$$Q(\tilde{P}, \theta) \;=\; \mathcal{L}(\theta) - KL(\tilde{P}||P(Z|X, \theta))$$

where $P(X, Z|\theta) = \prod_i \prod_k P[z_i = k]P[x_i|\theta_k]$
$\tilde{P}(Z)$ is any distribution over Z,
$KL(P(w)||Q(w)) = \sum_w P(w) \ln \frac{P(w)}{Q(w)}$ the **Kullbach-Leibler divergence**

Then,

  - **E step** $\max_{\tilde{P}} Q \Leftrightarrow \; KL(\tilde{P}||P(Z|X, \theta)$
  - **M step** $\max_\theta Q \Leftrightarrow \; KL(P(X|Z, \theta^{old})||P(X|\theta))$

- Interpretation: KL is "distance", "shortest distance" = projection

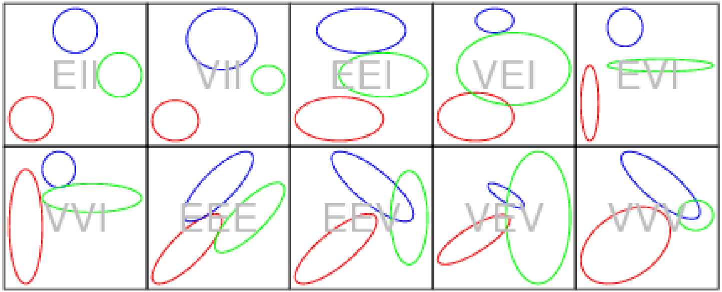
The M step in special cases

- Note that the expressions for μ_k, Σ_k = expressions for μ, Σ in the normal distribution, with data points x_i weighted by $\frac{\gamma_{ki}}{\Gamma_k}$

	M step
general case	$\Sigma_k = \sum_{i=1}^n \frac{\gamma_{ki}}{\Gamma_k}(x_i - \mu_k)(x_i - \mu_k)^T$
$\Sigma_k = \Sigma$ "same shape & size" clusters	$\Sigma \leftarrow \frac{\sum_{i=1}^n \sum_{k=1}^K \gamma_{ki}(x_i-\mu_k)(x_i-\mu_k)^T}{n}$
$\Sigma_k = \sigma_k^2 I_d$ "round" clusters	$\sigma_k^2 \leftarrow \frac{\sum_{i=1}^n \gamma_{ki}\|\|x_i-\mu_k\|\|^2}{d\Gamma_k}$
$\Sigma_k = \sigma^2 I_d$ "round, same size" clusters	$\sigma^2 \leftarrow \frac{\sum_{i=1}^n \sum_{k=1}^K \gamma_{ki}\|\|x_i-\mu_k\|\|^2}{nd}$

Exercise Prove the formulas above

- Note also that **K-means** is **EM** with $\Sigma_k = \sigma^2 I_d,\ \sigma^2 \to 0$ Exercise Prove it



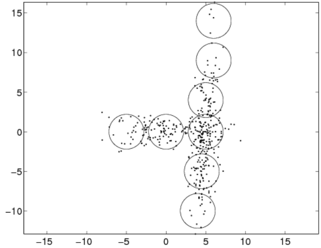
More special cases [?] introduce the following description for a covariance matrice in terms of *volume, shape, alignment with axes* (=determinant, trace, e-vectors). The letters below mean: I=unitary (shape, axes), E=equal (for all k), V=unequal

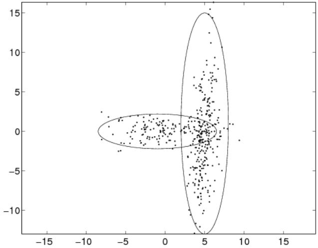
- EII: equal volume, round shape (spherical covariance)
- VII: varying volume, round shape (spherical covariance)
- EEI: equal volume, equal shape, axis parallel orientation (diagonal covariance)
- VEI: varying volume, equal shape, axis parallel orientation (diagonal covariance)
- EVI: equal volume, varying shape, axis parallel orientation (diagonal covariance)
- VVI: varying volume, varying shape, equal orientation (diagonal covariance)
- EEE: equal volume, equal shape, equal orientation (ellipsoidal covariance)
- EEV: equal volume, equal shape, varying orientation (ellipsoidal covariance)
- VEV: varying volume, equal shape, varying orientation (ellipsoidal covariance)
- VVV: varying volume, varying shape, varying orientation (ellipsoidal covariance)

(from [?])

EM versus K-means

- Alternates between cluster assignments and parameter estimation
- Cluster assignments γ_{ki} are probabilistic
- Cluster parametrization more flexible





- Converges to local optimum of **log-likelihood**
  Initialization recommended by K-logK method []

- **Modern algorithms with guarantees** (for e.g. mixtures of Gaussians)
  - Random projections
  - Projection on principal subspace [**?**]
  - Two step EM (=K-logK initialization + one more EM iteration) []

"Computer science" algorithms for mixture models

- Assume clusters well-separated **(S)**
  - e.g $||\mu_k - \mu_l|| \geq C \max(\sigma_k, \sigma_l)$
  - with $\sigma_k^2 = \max \mathrm{eigenvalue}(\Sigma_k)$
- true distribution is mixture
  - of Gaussians
  - of **log-concave** f_k's (i.e. $\ln f_k$ is concave function)
- then, w.h.p. (n, K, d, C)
  - we can label all data points correctly
  - $\Rightarrow$ we can find good estimate for θ

Even with **(S)** this is not an easy task in high dimensions
Because $f_k(\mu_k) \to 0$ in high dimensions (i.e there are few points from Gaussian k near μ_k)

The Vempala-Wang algorithm[?]

Idea

Let $\mathcal{H} = \mathrm{span}(\mu_{1:K})$
Projecting data on $\mathcal{H}$

- $\approx$ preserves $||x_i - x_j||$ if $k(i) \neq k(j)$
- $\approx$ reduces $||x_i - x_j||$ if $k(i) = k(j)$
- density at μ_k increases

(Proved by Vempala & Wang, 2004[?]) $\mathcal{H} \approx K$-th principal subspace of data

Algorithm Vempala-Wang (sketch)

1. Project points $\{x_i\} \in \mathbb{R}^d$ on $K-1$-th principal subspace $\Rightarrow \{y_i\} \in \mathbb{R}^K$
2. do distance-based "harvesting" of clusters in $\{y_i\}$

Other "CS" algorithms

- [?] round, equal sized Gaussian, random projection
- [?] arbitrary shaped Gaussian, distances
- [?] log-concave, principal subspace projection

Example Theorem (Achlioptas & McSherry, 2005) If data come from K Gaussians, $n >> K(d + \log K)/\pi_{min}$, and

$$||\mu_k - \mu_l|| \geq 4\sigma_k\sqrt{1/\pi_k + 1/\pi_l} + 4\sigma_k\sqrt{K\log nK + K^2}$$

then, w.h.p. $1 - \delta(d, K, n)$, their algorithm finds true labels

Good

- theoretical guarantees
- no local optima
- suggest heuritics for EM K-means
  - project data on principal subspace (when $d >> K$)

But

- strong assuptions: large separation (unrealistic), concentration of f_k's (or f_k known), K known
- try to find perfect solution (too ambitious)

A fundamental result

The Johnson-Lindenstrauss Lemma For any $\varepsilon \in (0,1]$ and any integer n, let d' be a positive integer such that $d' \geq 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \ln n$. Then for any set $\mathcal{D}$ of n points in $\mathbb{R}^d$, there is a map $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ such that for all $u, v \in V$,

$$(1-\varepsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (1+\varepsilon)||u-v||^2 \tag{14}$$

Furthermore, this map can be found in randomized polynomial time.

- note that the **embedding dimension** d' does not depend on the original dimension d, but depends on n, ε
- [?] show that: the mapping f is linear and that w.p. $1-\frac{1}{n}$ a random projection (rescaled) has this property
- their proof is elementary Projecting a fixed vector v on a a random subspace is the same as projecting a random vector v on a fixed subspace. Assume $v = [v_1, \ldots v_d]$ with $v \sim$ i.i.d. and let $\tilde{v}$ = projection of v on axes $1 : d'$. Then $E[||\tilde{v}||^2 = d' E[v_j^2] = \frac{d'}{d} E[||v||^2]$. The next step is to show that the variance of $||\tilde{v}||^2$ is very small when d' is sufficiently large.

A two-step EM algorithm [?]

Assumes K spherical gaussians, separation $||\mu_k^{true} - \mu_{k'}^{true} \geq C\sqrt{d}\sigma_k$

1. Pick $K' = \mathcal{O}(K \ln K)$ centers μ_k^0 at random from the data
2. Set $\sigma_k^0 = \frac{d}{2} \min_{k \neq k'} ||\mu_k^0 - \mu_{k'}^0||^2$, $\pi_k^0 = 1/K'$
3. Run one E step and one M step $\Longrightarrow \{\pi_k^1, \mu_k^1, \sigma_k^1\}_{k=1:K'}$
4. Compute "distances" $d(\mu_k^1, \mu_{k'}^1) = \frac{||\mu_k^1 - \mu_{k'}^1||}{\sigma_k^1 - \sigma_{k'}^1}$
5. Prune all clusters with $\pi_k^1 \leq 1/4K'$
6. Run Fastest First Traversal with distances $d(\mu_k^1, \mu_{k'}^1)$ to select K of the remaining centers. Set $\pi_k^1 = 1/K$.
7. Run one E step and one M step $\Longrightarrow \{\pi_k^2, \mu_k^2, \sigma_k^2\}_{k=1:K}$

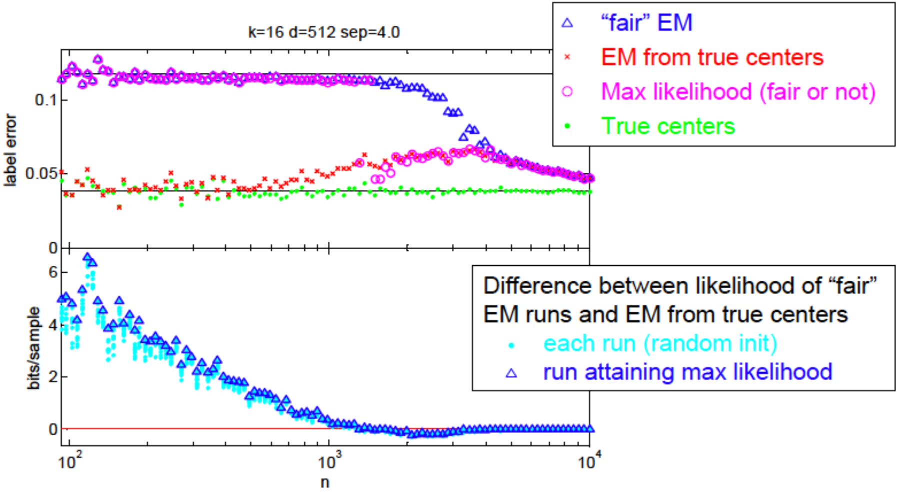
eorem For any $\delta, \varepsilon > 0$ if d large, n large enough, separation $C \geq d^{1/4}$ the **Two step EM** algorithm obtains centers μ_k so that

$$||\mu_k - \mu_k^{true}|| \leq ||\mathrm{mean}(C_k^{true}) - \mu_k^{true}|| + \varepsilon\sigma_k\sqrt{d}$$

Experimental exploration [?]

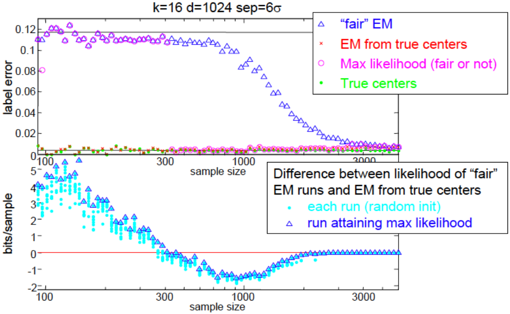
- High d
- True model: centers μ_k^* at corners of hypercube, $\Sigma_k^* = \sigma I_d$ spherical equal covariances, $\pi_k^* = 1/K$
- n, K, separation variable
- Algorithm: EM with Power initialization and projection on $(K-1)$-th principal subspace

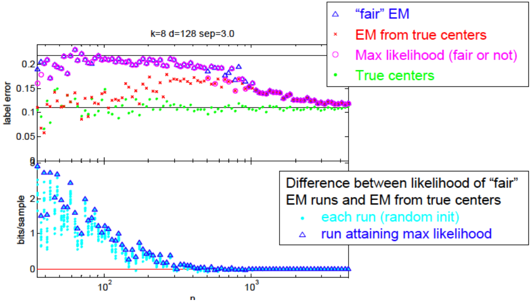
Experimental exploration [?] (2)



figures from [?]

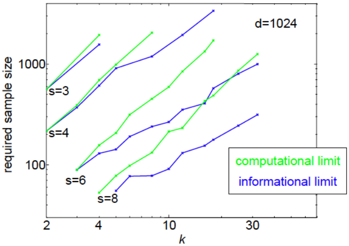
Experimental exploration [?] (3)





Experimental exploration [?] (4)

- Practical limits vs theoretical limits



$n \propto k^{1.5} - k^{1.6}$ for all d, separation

figures from [?]

Dasgupta 1999	$s > 0.5d^{1/2}$	$n = \Omega(k^{\log^2 1/\delta})$	Random projection, then mode finding
Dagupta Schulamn 2000	$s = \Omega(d^{1/4})$ (large d)	n = poly(k)	2 round EM with $\Theta(k \cdot \log k)$ centers
Arora Kannan 2001	$s = \Omega(d^{1/4} \log d)$		Distance based
Vempala Wang 2004	$s = \Omega(k^{1/4} \log dk)$	$n = \Omega(d^3k^2\log(dk/s\delta))$	Spectral projection, then distances

General mixture of Gaussians:

[Kannan Salmasian Vempala 2005] $s=\Omega(k^{5/2}\log(kd))$, $n=\Omega(k^2d \cdot \log^5(d))$

[Achliopts McSherry 2005] $s>4k+o(k)$, $n=\Omega(k^2d)$

Selecting K

- Run clustering algorithm for $K = K_{min} : K_{max}$
  - obtain $\Delta_{K_{min}}, \ldots \Delta_{K_{max}}$ or $\gamma_{K_{min}}, \ldots \gamma_{K_{max}}$
  - choose best Δ_K (or γ_K) from among them
- Typically increasing $K \Rightarrow$ cost $\mathcal{L}$ decreases
  - ($\mathcal{L}$ cannot be used to select K)
  - Need to "penalize" $\mathcal{L}$ with function of number parameters

Selecting K for mixture models

The **BIC (Bayesian Information) Criterion**

- let θ_K = parameters for γ_K
- let $\#\theta_K$=number independent parameters in θ_K
  - e.g for mixture of Gaussians with full Σ_k's in d dimensions

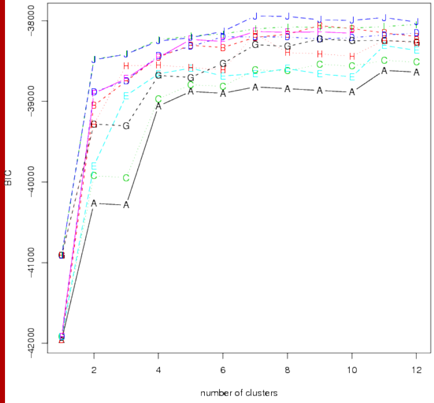
$$\#\theta_K = \underbrace{K-1}_{\pi_{1:K}} + \underbrace{Kd}_{\mu_{1:K}} + \underbrace{Kd(d-1)/2}_{\Sigma_{1:K}}$$

- define

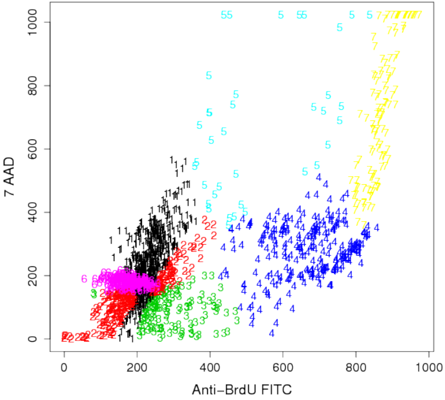
$$BIC(\theta_K) \;=\; l(\theta_K) - \frac{\#\theta_K}{2} \ln n$$

- **Select** K that maximizes $BIC(\theta_K)$
- selects true K for $n \to \infty$ and other technical conditions (e.g parameters in compact set)
- but theoretically not justified (and overpenalizing) for finite n

Number of Clusters vs. BIC EII (A), VII (B), EEI (C), VEI (D), EVI (E), VVI (F), EEE (G), EEV (H), VEV (I), VVV (J)

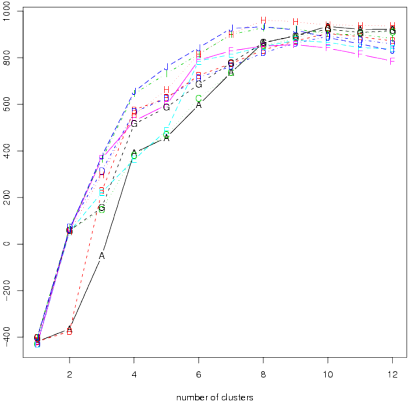


EEV, 8 Cluster Solution

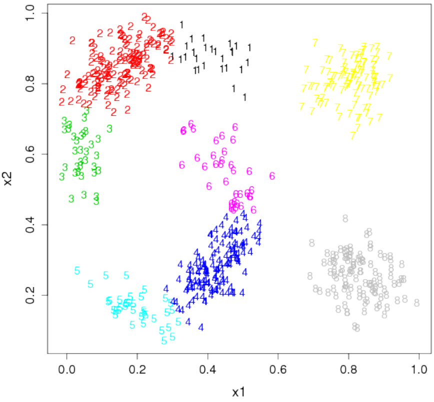


(from [?])

Number of Clusters vs. BIC EII (A), VII (B), EEI (C), VEI (D), EVI (E), VVI (F), EEE (G), EEV (H), VEV (I), VVV (J)



EEV, 8 Cluster Solution



(from [?])