Lecture VII: Clustering: K-means and Mixtures of Gaussians

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

March, 2026

Paradigms for clustering

K-means clustering

Mixtures of Gaussians and the EM algorithm

Special topics in clustering

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

What is clustering? Problem and Notation

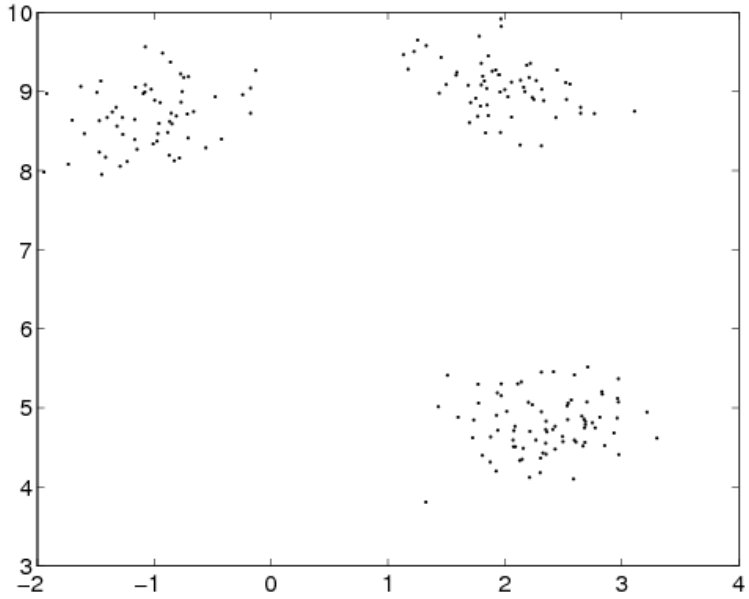
- **Informal definition** **Clustering** = Finding groups in data
- **Notation**

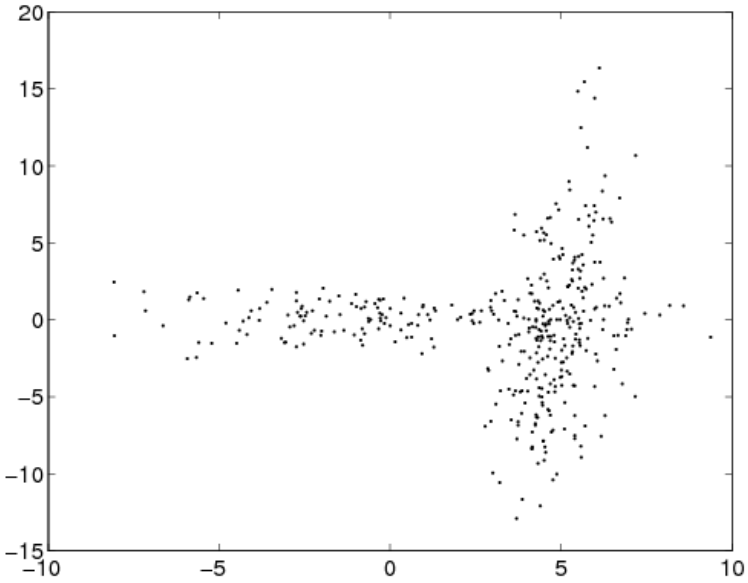
$\mathcal{D}$	=	$\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$ a **data set**
n	=	number of **data points**
K	=	number of **clusters** ($K << n$)
Δ	=	$\{C_1, C_2, \ldots, C_K\}$ a partition of $\mathcal{D}$ into disjoint subsets
$k(i)$	=	the **label** of point i
$\mathcal{L}(\Delta)$	=	cost (loss) of Δ (to be minimized)

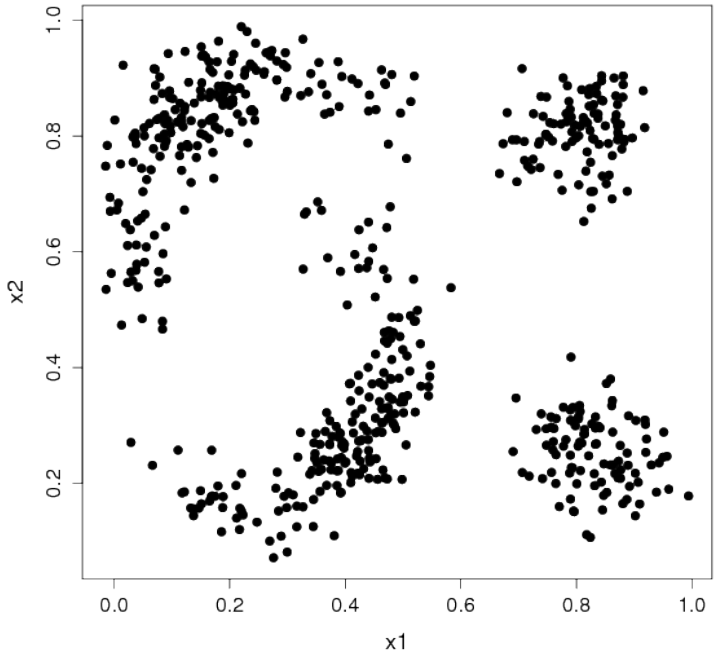
- **Second informal definition** **Clustering** = given n **data points**, separate them into K **clusters**
- Hard vs. soft clusterings
  - **Hard** clustering Δ: an item belongs to only 1 cluster
  - **Soft** clustering $\gamma = \{\gamma_{ki}\}_{k=1:K}^{i=1:n}$
    γ_{ki} = the **degree of membership**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

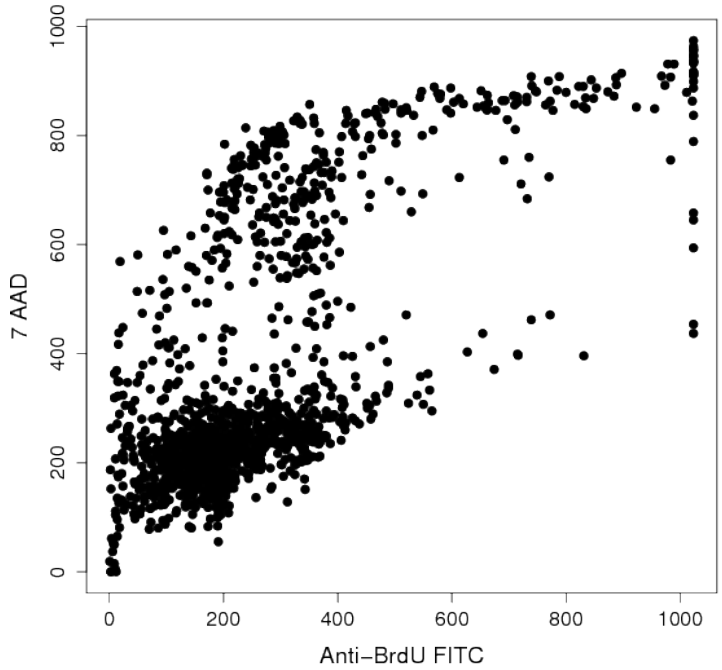
$$\sum_k \gamma_{ki} = 1 \quad \text{for all } i$$

(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	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]

- 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

K-means clustering

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)

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

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

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

- 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

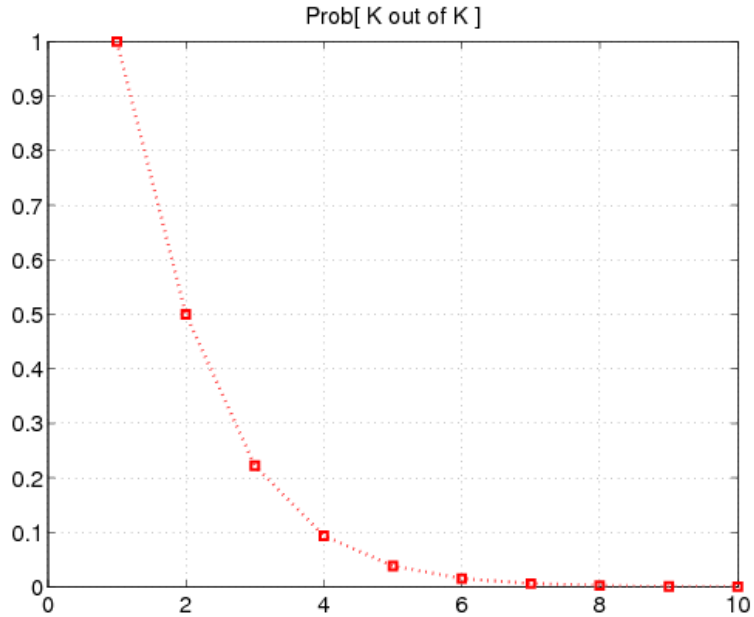
$$\mathcal{L}(\Delta) = -\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \notin C_k} ||x_i - x_j||^2 + \text{constant} \tag{5}$$

Proof of (4) Replace μ_k as expressed in (1) in the expression of $\mathcal{L}$, then rearrange the terms

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

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++** Like FFT but probabilistic; μ_{k+1} is point i w.p. $\propto \min_{k'=1:k} \|x^i - \mu_{k'}\|^2$
- 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

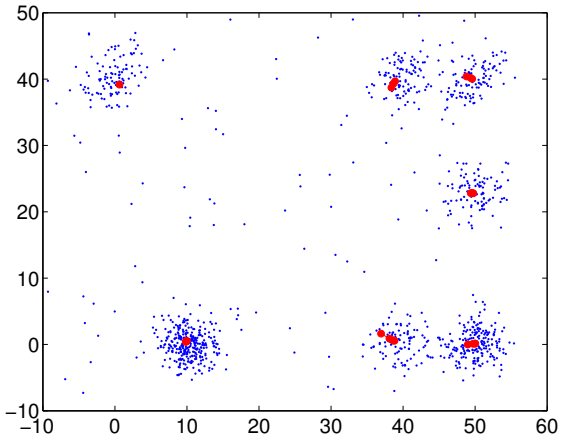
The K-logK Initialization (see also)

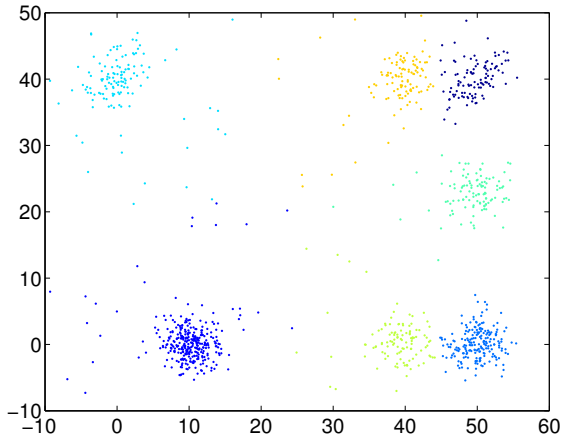
1. pick $\mu^0_{1:K'}$ at random from data set, where $K' = O(K \log K)$
   (this assures that each cluster has at least 1 center w.h.p)
2. run 1 step of K-means
3. remove all centers μ^0_k that have few points, e.g $|C_k| < \frac{n}{eK'}$
4. from the remaining centers select K centers by **Fastest First 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

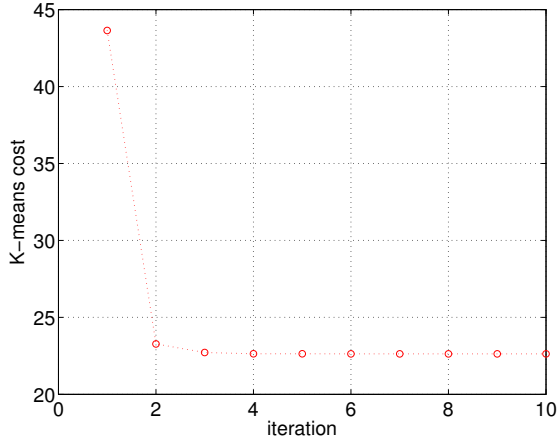
K-means clustering with K-logK Initialization

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

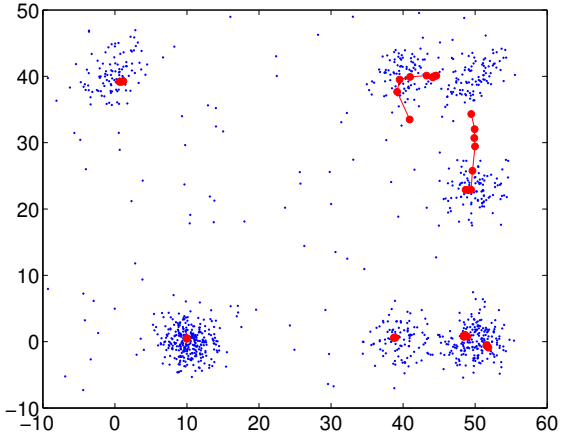
K-LOGK $K = 7$, $T = 100$, $n = 1100$, $c = 1$

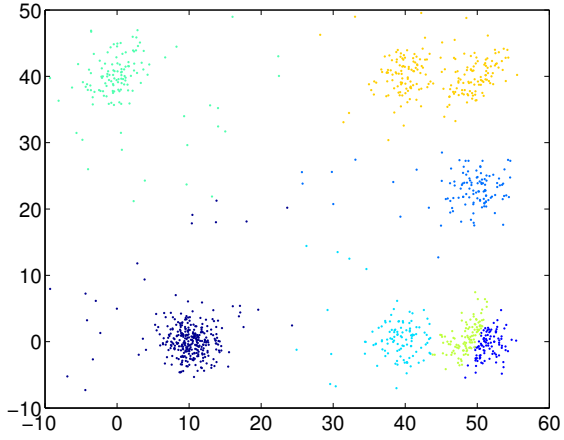


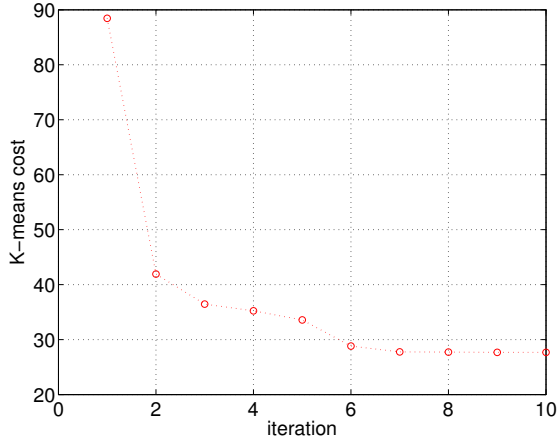




NAIVE $K = 7$ $T = 100$, $n = 1100$







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 $k = 2 : K$

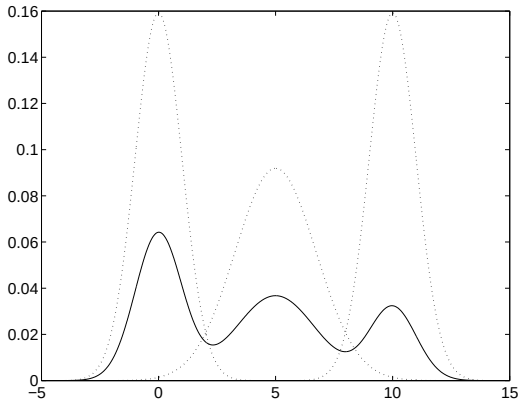
$$\mu_k \leftarrow \operatorname*{argmax}_{\mathcal{D}} \text{distance}(x_i, \{\mu_{1:k-1}\})$$

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

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

Model based clustering: Mixture models

Mixture in 1D



- The **mixture density**

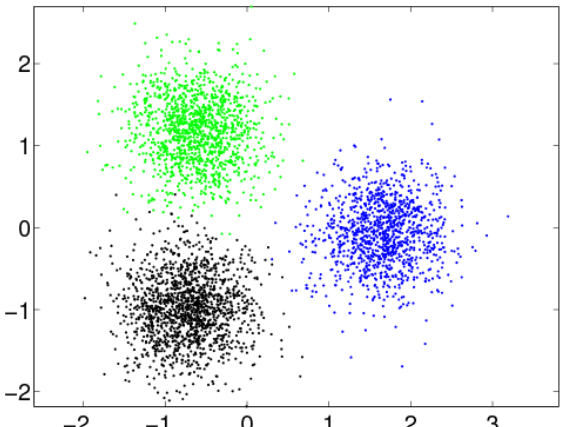
$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x)$$

- $f_k(x)$ = 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

Mixture in 2D



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 $i = 1 : n$, $k = 1 : K$

$$\gamma_{ki} = \frac{\pi_k f_k(x)}{f(x)} \tag{10}$$

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 (14)
- $\sum_k \Gamma_k = n$

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

The EM Algorithm – Motivation

- Define the **indicator variables**

$$z_{ik} = \begin{cases} 1 & \text{if } i \in C_k \\ 0 & \text{if } i \notin C_k \end{cases} \tag{11}$$

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

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

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

$$E[l_c(\theta, \bar{z})] = \sum_{i=1}^{n} \sum_{k=1}^{K} E[z_{ki}][\ln \pi_k + \ln f_k(x_i)] \tag{13}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ki} \ln \pi_k + \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ki} \ln f_k(x_i)] \tag{14}$$

- 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

$$Q(\theta, \gamma) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ki} \ln \underbrace{\pi_k f_k(x_i)}_{\theta} \tag{15}$$

- 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

$$Q(\tilde{P}, \theta) = \mathcal{L}(\theta) - KL(\tilde{P}||P(Z|X, \theta) \tag{16}$$

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

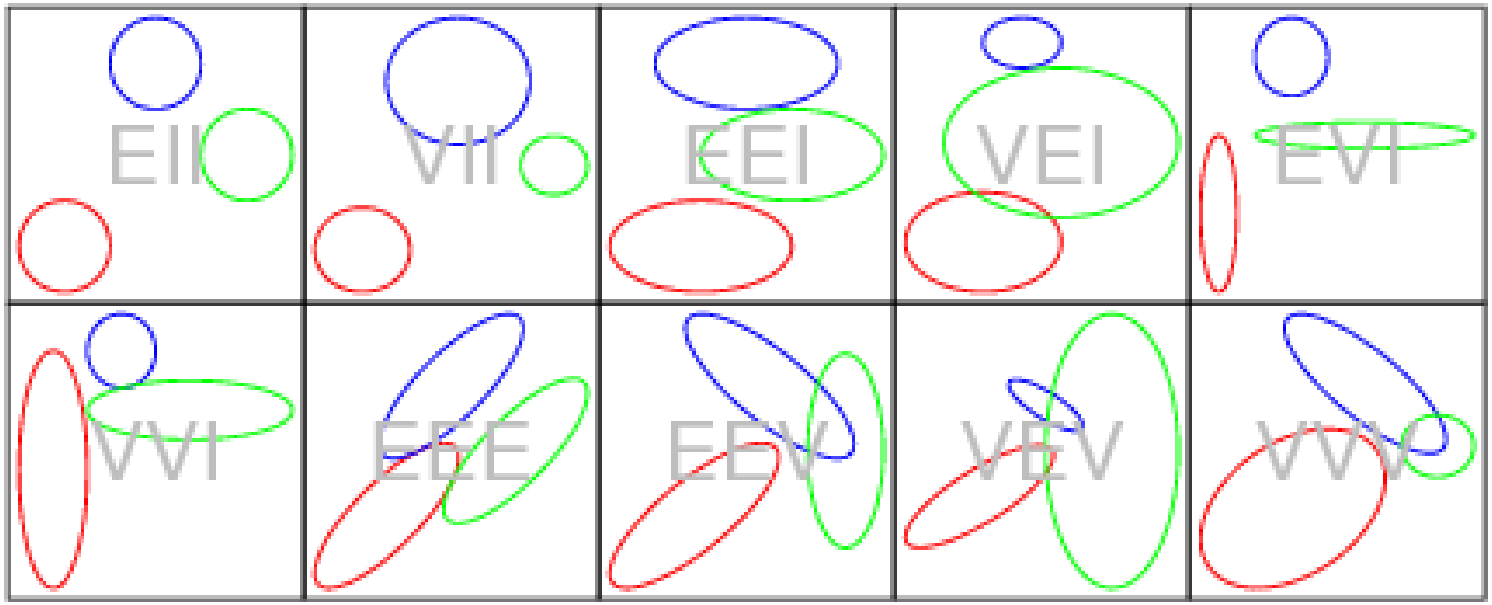
The M step in special cases

- 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



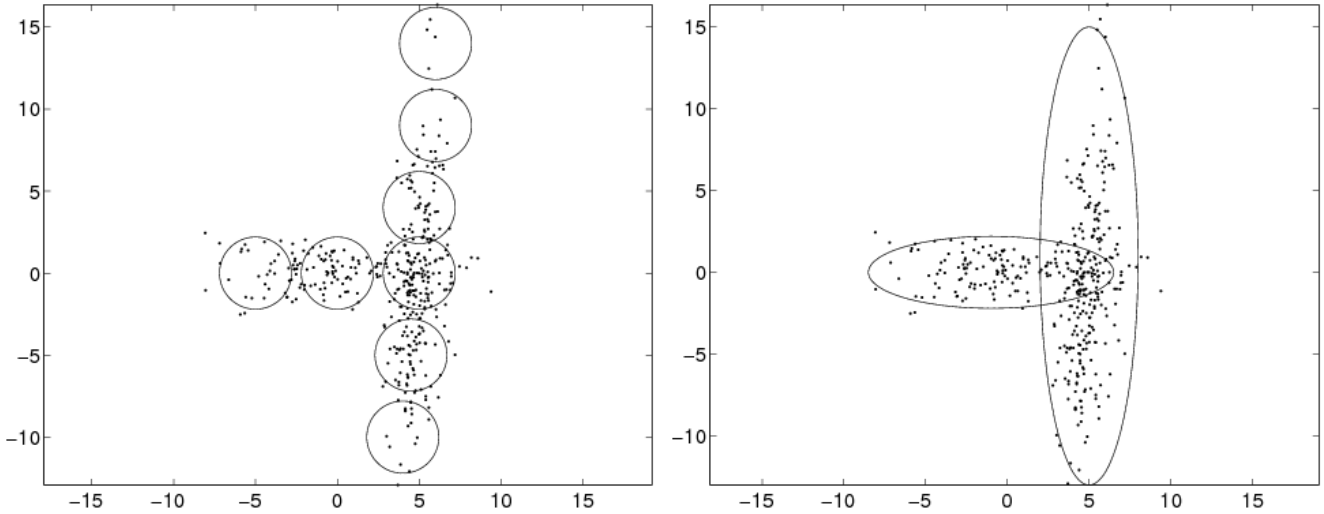
More special cases introduce the following description for a covariance matrice in terms of *volume, shape, alignment with axes* (=determinant, trace, e-vectors). The letters below mean: I=unitary (shape, axes), E=equal (for all k), V=unequal

- 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)

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 $u, v \in V$,

$$(1 - \varepsilon)||u - v||^2 \leq ||f(u) - f(v)||^2 \leq (1 + \varepsilon)||u - v||^2 \tag{17}$$

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

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

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

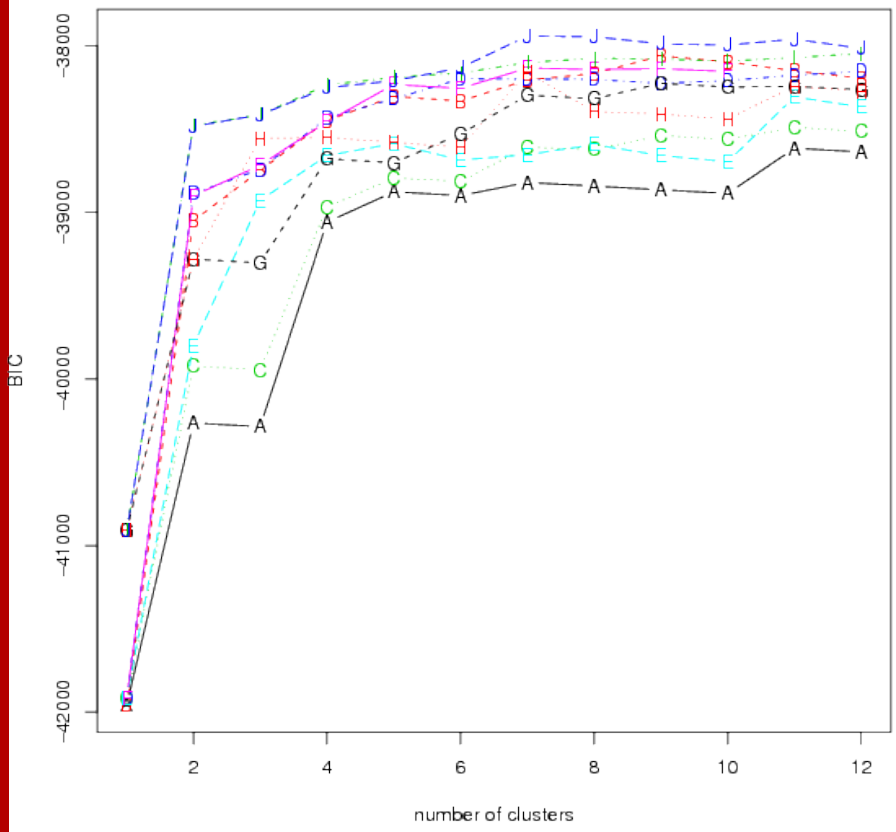
$$\#\theta_K = \underbrace{K-1}_{\pi_{1:K}} + \underbrace{Kd}_{\mu_{1:K}} + \underbrace{Kd(d-1)/2}_{\Sigma_{1:K}}$$

- define

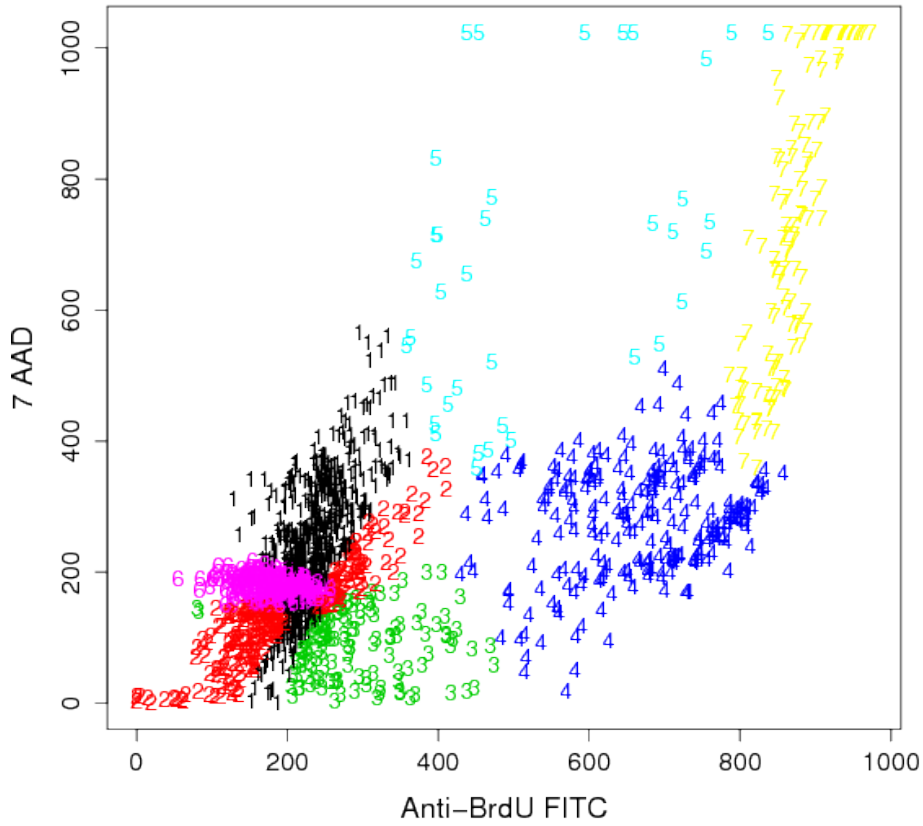
$$BIC(\theta_K) = l(\theta_K) - \frac{\#\theta_K}{2} \ln n$$

- **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)

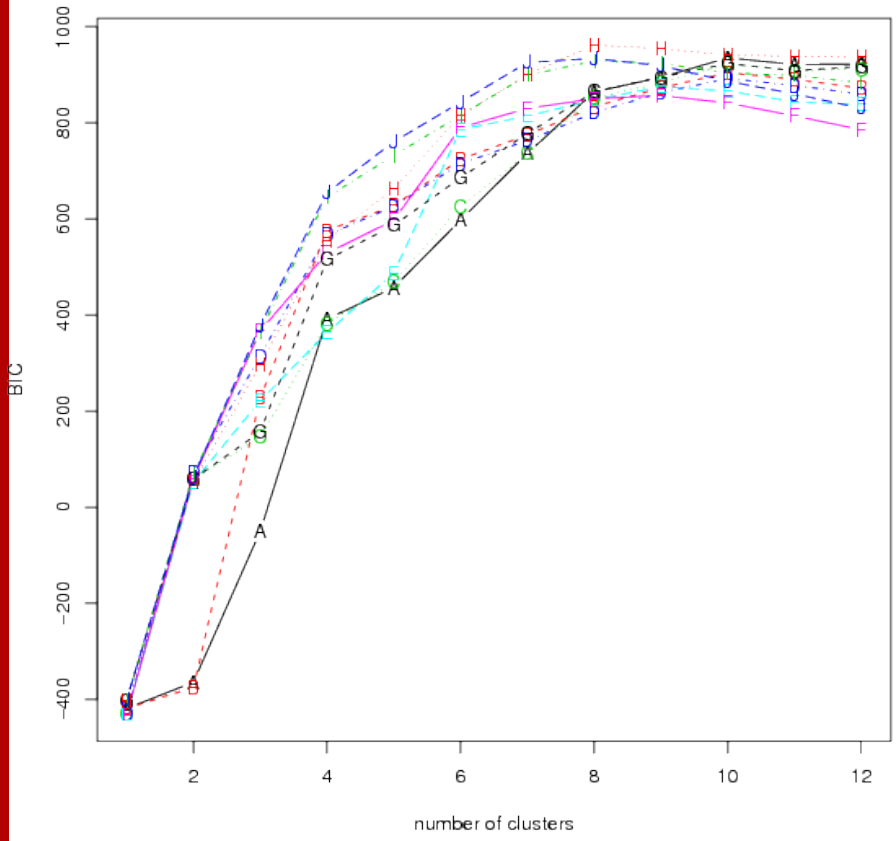


EEV, 8 Cluster Solution

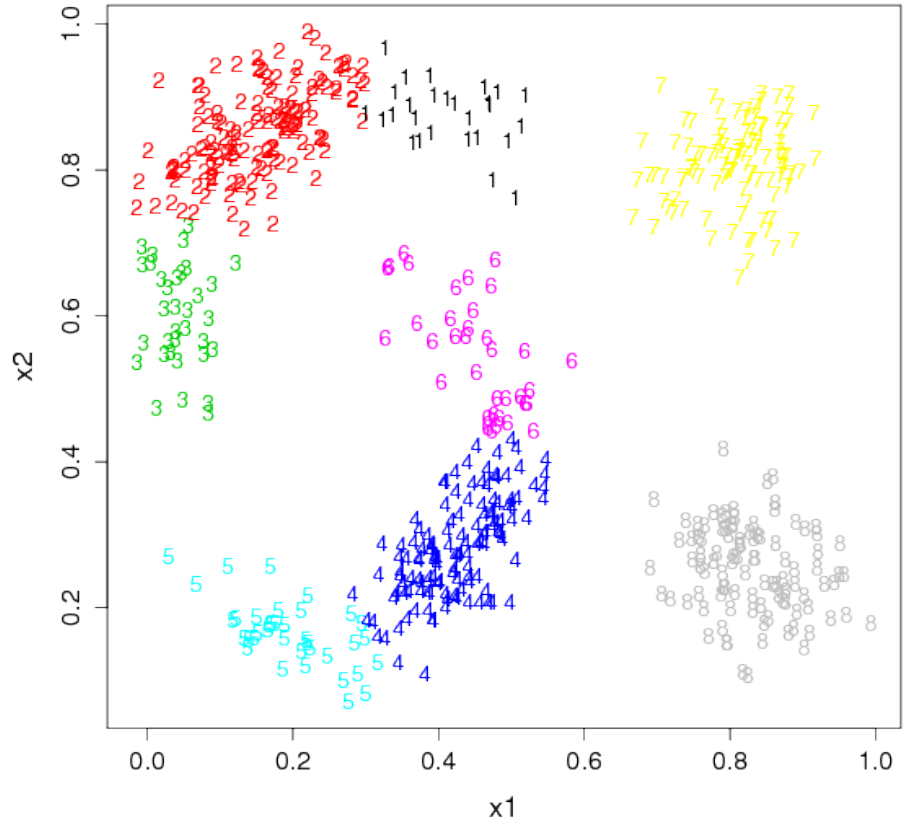


(from)

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)

Selecting K for hard clusterings

- based on statistical testing: the **gap** statistic (Tibshirani, Walther, Hastie, 2000)
- **X-means** heuristic: splits/merges clusters based on statistical tests of Gaussianity
- Stability methods
  - Empirical – prove instability
  - Optimization based – prove stability

Empirical Stability methods for choosing K

- like bootstrap, or crossvalidation
- **Idea** (implemented by)

  for each K
  1. perturb data $\mathcal{D} \rightarrow \mathcal{D}'$
  2. cluster $\mathcal{D}' \rightarrow \Delta'_K$
  3. compare Δ_K, Δ'_K. Are they similar?
     If yes, we say Δ_K is **stable to perturbations**

Fundamental assumption If Δ_K is stable to perturbations then K is the correct number of clusters

- these methods are supported by experiments (not extensive)
- **not directly supported by theory** . . . see for a summary of the area

- And some theory: rigurous positive results + limitations of empirical stability methods

How to tell when a clustering is (approximately) correct using convex relaxations

Marina Meila*
University of Washington
Seattle, WA 98195
mmp@stat.washington.edu

A Sober Look at Clustering Stability

Shai Ben-David[1], Ulrike von Luxburg[2], and Dávid Pál[1]

[1] David R. Cheriton School of Computer Science,
University of Waterloo,
Waterloo, Ontario, Canada
{shai,dpal}@cs.uwaterloo.ca
[2] Fraunhofer IPSI, Darmstadt, Germany
ulrike.luxburg@ipsi.fraunhofer.de

What I didn't talk about

- Hierarchical clustering
- Subspace clustering (or clustering on subsets of attributes)
- Bi-clustering (and multi-way-clustering)
- Partial clustering
- Non-parametric clustering
- Ensembles of clusterings, consensus clustering, and clustering clusterings

Hierarchical clustering

- **Divisive** (top down)
  - starts with all data in one cluster, divides recursively into 2 (or more) clusters
  - Example: spectral clustering, min diameter
- **Agglomerative** (bottom up)
  - starts n cluster containing 1 item, merges 2 clusters recursively
  - Example: Ward algorithm, single linkage
- **Hierarchical Dirichlet processes**
- Remarks
  - Any cost based clustering paradigm can produce a hierarchical clustering
  - Any non-parametric level-sets paradigm can produce a hierarchical clustering
  - Mixture models (finite or not) can also be defined hierarchically. Issues of identifyability appear

The Ward agglomerative algoritm

- Cost = same as K-means
- Algorithm idea:
  - Start with n single point clusters
  - Merge the two clusters that increase $\mathcal{L}$ the least, until K clusters left
- **Greedy**, recursive algorithm, $\mathcal{O}(n^3)$ operations

Subspace clustering

- Problem: each cluster is defined by a subset of relevant attributes (features)
  - Examples: user modeling (clusters of users vs clusters of products/services), gene expression data
- Known as **Clustering on Subsets of Attributes (COSA) Biclustering (and Multiway Clustering), Subspace clustering**
- Amounts to clustering both the data exemplars and the data features
- Approaches
  - **COSA** cost based, $+$ additional entropy term. Alternate minimization algorithm.
  - Dirichlet process mixtures approach. Each $f(.;\theta_k)$ samples a set of relevant features. Estimated by MCMC
  - **Multivariate Information Bottleneck** Information theory based. Estimation by alternate (KL-divergence) projections.
  - many others. . . see IEEE TKDE

Partial clustering

- Problem: Given a node, find its cluster
- Premise: the data set is extremely large, there are many small clusters, possibly $\mathcal{O}(n)$
- **Nibble** algorithm of

  Given: a graph, by its Markov transition matrix P
  Start with node i, tolerance ε, number steps t
  Initialize $p \in \mathbb{R}^n$ with $p_i = 1$, $p_j = 0$ for $j \neq i$
  - Iterate for t steps
    1. $p \leftarrow Pp$
    2. for $j = 1 : n$, if $p_j < \varepsilon$ set $p_j = 0$

  Output $C(i) = \{ j \mid p_j > 0 \}$
- $C(i)$ is the set of items attainable from i by a "likely" path
- Original algorithm has sparsest cut guarantees
  Used as subroutine by other algorithms.

Methods based on non-parametric density estimation

Idea The clusters are the isolated peaks in the (empirical) data density

- group points by the peak they are under
- some outliers possible
- $K = 1$ possible(no clusters)
- shape and number of clusters K determined by algorithm
- **structural parameters**
  - **smoothness** of the **density estimate**
  - what is a peak

Algorithms

- peak finding algorithms Mean-shift algorithms
- level sets based algorithms
  - Nugent-Stuetzle, Support Vector clustering
- Information Bottleneck