BIOSTAT 527 4/24/23

Lecture 9

Poll → Topics

Clustering

SVM → Kernel Machines

Cost based - "K-means"

Mixture models

Hierarchical

K clusters paradigms

(NP, K = 1 : n)

[Comparing clusterings
Cluster[ing] validation] Brief

next time

Lecture VII: Classic and Modern Data Clustering – Part I

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

November, 2020

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

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)

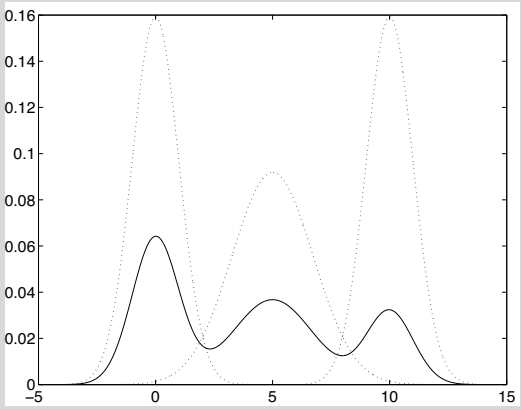
- **Second informal definition** **Clustering** = given n **data points**, separate them into K **clusters**
- Hard vs. soft clusterings
  - **Hard** clustering Δ: an item belongs to only 1 cluster
  - **Soft** clustering $\gamma = \{\gamma_{ki}\}_{k=1:K}^{i=1:n}$
    γ_{ki} = the **degree of membership**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

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

(usually associated with a probabilistic model)

Model based clustering: Mixture models

Mixture in 1D



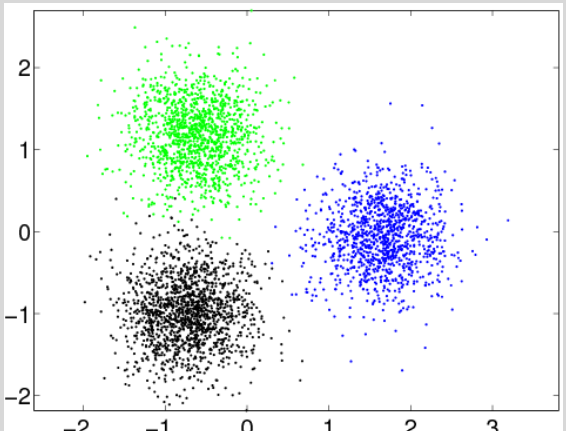
- The **mixture density**

weighted sum

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

- $f_k(x)$ = 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** $\theta = (\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K})$

Mixture in 2D



- The **degree of membership** of point i to cluster k

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

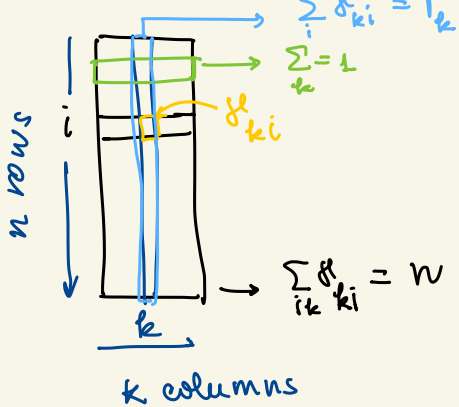
Clustering

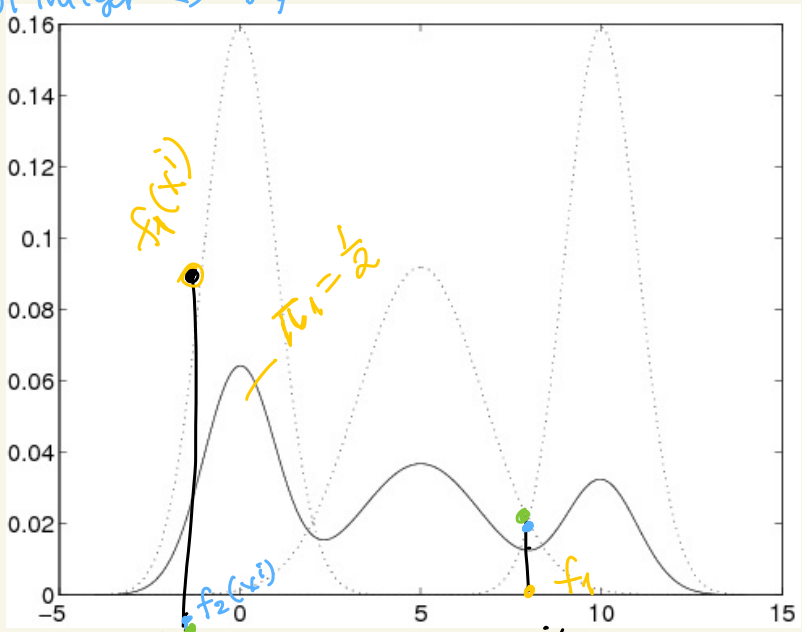
- depends on x_i and on the model parameters

probabilistic

$$\mu_{ki} = \Pr[k(i) = k \mid x^i] = \frac{\pi_k f_k(x^i)}{\sum_{k'} \pi_{k'} f_{k'}(x^i)} \Rightarrow \sum_{k=1}^{K} \mu_k(i) = 1$$

μ_{ki} or $\mu_k(i)$ — $k(i)$: cluster of i, $\in 1:K$





$\pi_1 = 0.5 \quad \pi_3 = \frac{1}{6}$

$\pi_2 = 1/3$

$\mu_{1i} = \frac{0.5 \cdot 1}{0.5 \cdot 1 + \frac{1}{3}\varepsilon_2 + \frac{1}{6}\cdot\varepsilon_3} \approx 1$

$\mu_1(x^{i'}) \approx 0$

$\mu_2(x^{i'}) \simeq \frac{\frac{1}{3} \cdot 1}{\frac{1}{3} \cdot 1 + \frac{1}{6} \cdot 1} \simeq \frac{2}{3}$

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**

maximize $$l(\theta) = \ln \prod_{i=1}^{n} f(x_i) = \sum_{i=1}^{n} \ln \underbrace{\sum_{k} \pi_k f_k(x_i)}_{f(x_i)} \quad (9)$$

NO CLOSED FORM

- 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

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

How?

Given $\mathcal{D} = \{x^{1:n}\}$

Choose K, model class

Want $\underbrace{\pi_{1:K}, f_{1:K}}_{\text{parameters}}$

$\{g_k(x^{1:n}), k = 1:K\}$

Algorithms for model-based clustering

$\ell(\theta)$ = log likelihood

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

< guarantees depend on strong assumptions

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

Converges to local maximum
⇒ initialization important *
restarts

The Expectation-Maximization (EM) Algorithm

Alternate Optimization

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

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

assign $x^{1:n}$ to clusters

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

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

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

weighted mean

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

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

Converges to local max of $l(\theta)$

[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 $\theta \longrightarrow \theta^{ML}$ [McLachlan and Krishnan, 1997]
- the E and M steps can be seen as projections [Neal and Hinton, 1998]

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

Probablistic alternate projection view of EM[Neal and Hinton, 1998]

- 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

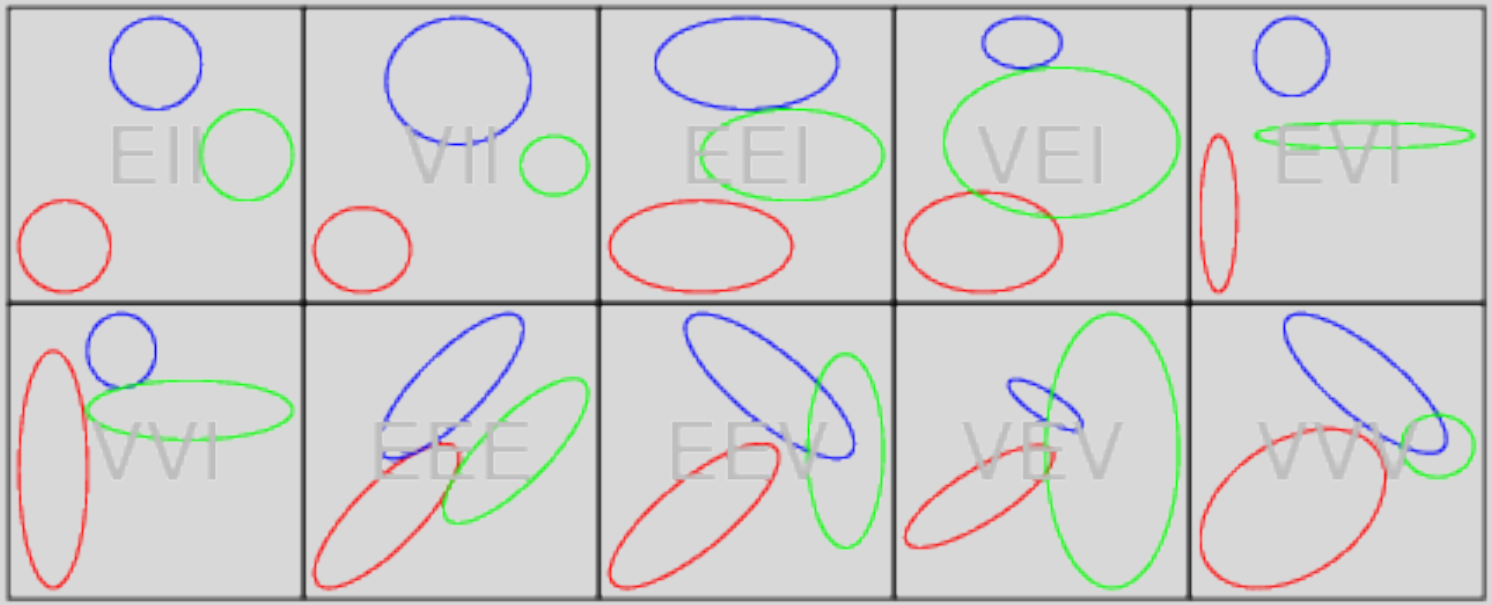
The M step in special cases

- 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



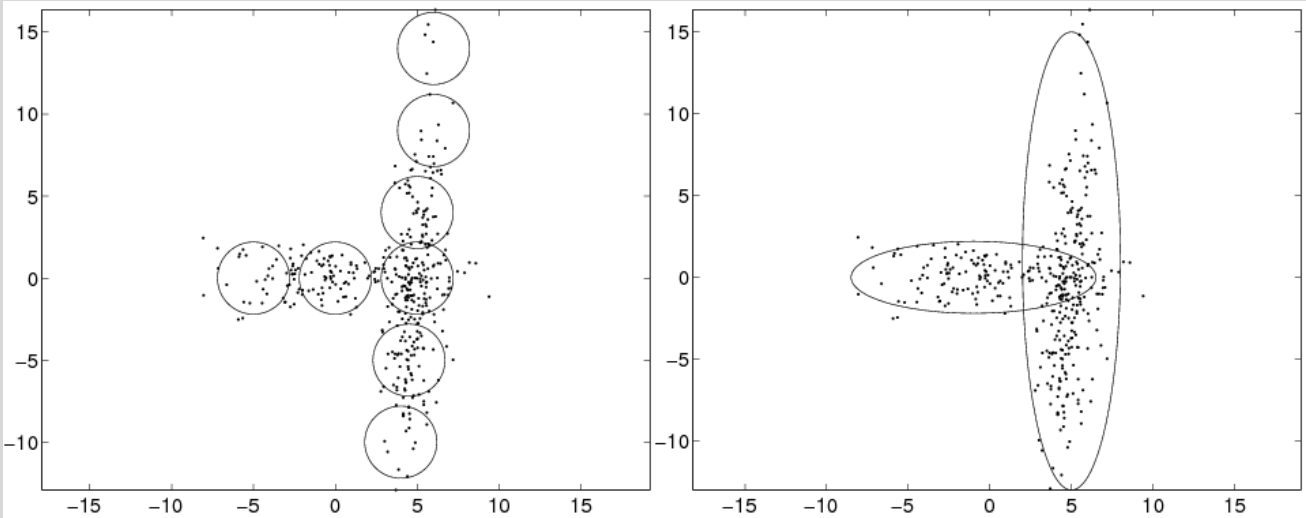
More special cases [Banfield and Raftery, 1993] introduce the following description for a covariance matrice in terms of *volume, shape, alignment with axes* (=determinant, trace, e-vectors). The letters below mean: I=unitary (shape, axes), E=equal (for all k), V=unequal

- 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 [Nugent and Meila, 2010])

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 [Vempala and Wang, 2004]
  - Two step EM (=K-logK initialization + one more EM iteration) []

[Parametric]

Paradigms → Algorithms

$\mathcal{L}$ = Cost : k-means L.S. — k-means

Model ⇔ $\mathcal{L}$ = -ln likelihood — EM, Gradient ascent, ...

Mixture models

← understand what clusters

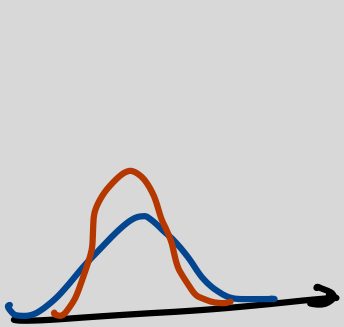
"Computer science" algorithms for mixture models

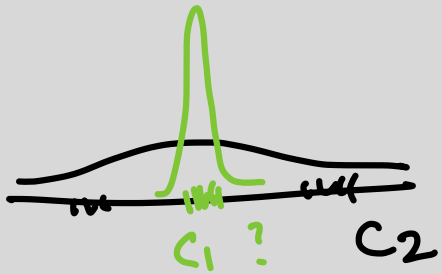
- Assume clusters well-separated
  - e.g $||\mu_k - \mu_l|| \geq C \max(\sigma_k, \sigma_l)$
  - with $\sigma_k^2 = \max \text{ eigenvalue}(\Sigma_k)$

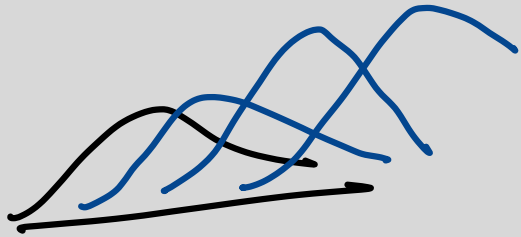
$\min_k \pi_k$ not too small (S)

- 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)







by $\hat{f}_k(i)$

Mixture but no clusters

The Vempala-Wang algorithm[Vempala and Wang, 2004]

Idea

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

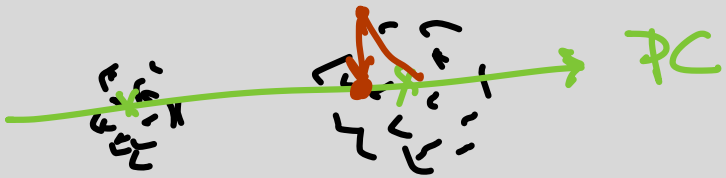
- $\approx$ preserves $||x_i - x_j||$ if $k(i) \neq k(j)$
- $\approx$ reduces $||x_i - x_j||$ if $k(i) = k(j)$
- density at μ_k increases

(Proved by Vempala & Wang, 2004[Vempala and 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 $\{y_i\}$

K centers → K-1 subspace



- dim ↘
- $d(x^i, \mu_k)$ ↘ more clustered
- more Gaussian (if $d \gg k$)

(μ_k: true center)

Other "CS" algorithms

- [Dasgupta, 2000] round, equal sized Gaussian, random projection
- [Arora and Kannan, 2001] arbitrary shaped Gaussian, distances
- [Achlioptas and McSherry, 2005] 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
- [Dasgupta and Gupta, 2002] 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 [Dasgupta and Schulman, 2007]

K-log K init

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

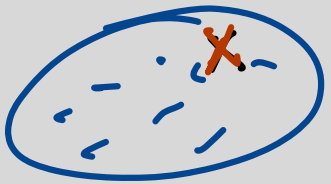
1. Pick $K' = \mathcal{O}(K \ln K)$ centers μ_k^0 at random from the data
2. Set $\sigma_k^0 = \frac{d}{2} \min_{k \neq k'} ||\mu_k^0 - \mu_{k'}^0||^2$, $\pi_k^0 = 1/K'$
3. Run one E step and one M step $\Longrightarrow \{\pi_k^1, \mu_k^1, \sigma_k^1\}_{k=1:K'}$
4. Compute "distances" $d(\mu_k^1, \mu_{k'}^1) = \frac{||\mu_k^1 - \mu_{k'}^1||}{\sigma_k^1 - \sigma_{k'}^1}$
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

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

Initialization more important than EM !!

when clusters good




Achlioptas, D. and McSherry, F. (2005).
On spectral learning of mixtures of distributions.
In Auer, P. and Meir, R., editors, *18th Annual Conference on Learning Theory, COLT 2005*, pages 458–471, Berlin/Heidelberg. Springer.


Arora, S. and Kannan, R. (2001).
Learning mixtures of arbitrary gaussians.
In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 247–257, New York, NY, USA. ACM Press.


Banfield, J. D. and Raftery, A. E. (1993).
Model-based gaussian and non-gaussian clustering.
Biometrics, 49:803–821.


Bradley, P. and Mangasarian, O. (2005).
Clustering via concave minimization.
In *Advances in Neural Information Processing systems (NIPS)*, Cambridge, MA. MIT Press.


Bubeck, S., Meilă, M., and von Luxburg, U. (2009).
How the initialization affects the stability of the k-means algorithm.
Technical Report arXiv:0907.5494v1 [stat.ML], ArXiv.


Carreira-Perpinan, M. A. (2007).
Gaussian mean shift is an EM algorithm.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 29(5):767–776.


Dasgupta, S. (2000).
Experiments with random projection.

In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 143–151, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.


Dasgupta, S. and Gupta, A. (2002).
An elementary proof of a theorem of johnson and lindenstrauss.
Algorithms, 22:60–65.


Dasgupta, S. and Schulman, L. (2007).
A probabilistic analysis of em for mixtures of separated, spherical gaussians.
Journal of Machine Learnig Research, 8:203–226.


Har-Peled, S. and Mazumdar, S. (2004).
Coresets for k-means and k-median clustering and their applications.
In *Proc. 36th Annu. ACM Sympos. Theory Comput (STOC)*, pages 291–300.


Hochbaum, D. S. and Shmoys, D. B. (1985).
A best possible heuristic for the k-center problem.
Mathematics of Operations Research, 10(2):180–184.


Lloyd, S. P. (1982).
Least squares quantization in PCM.
IEEE Transactions on Information Theory, 28:129–137.


McLachlan, G. J. and Krishnan, T. (1997).
The EM algorithm and extensions.
Wiley, New York, NY.


Neal, R. M. and Hinton, G. E. (1998).
A view of the em algorithm that justifies incremental, sparse, and other variants.
In Jordan, M. I., editor, *Learning in Graphical Models*, NATO Science series, pages 355–368. Kluwer Academic Publishers.


Nugent, R. and Meila, M. (2010).
Statistical Methods in Molecular Biology, chapter An Overview of Clustering Applied to Molecular Biology.
Humana Press, Springer.


Srebro, N., Shakhnarovich, G., and Roweis, S. (2006).
An investigation of computational and informational limits in gaussian mixture clustering.
In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*.


Vempala, S. and Wang, G. (2004).
A spectral algorithm for learning mixtures of distributions.
Journal of Computer Systems Science, 68(4):841–860.

Lecture IV – Hierarchical clustering. Comparing clusterings

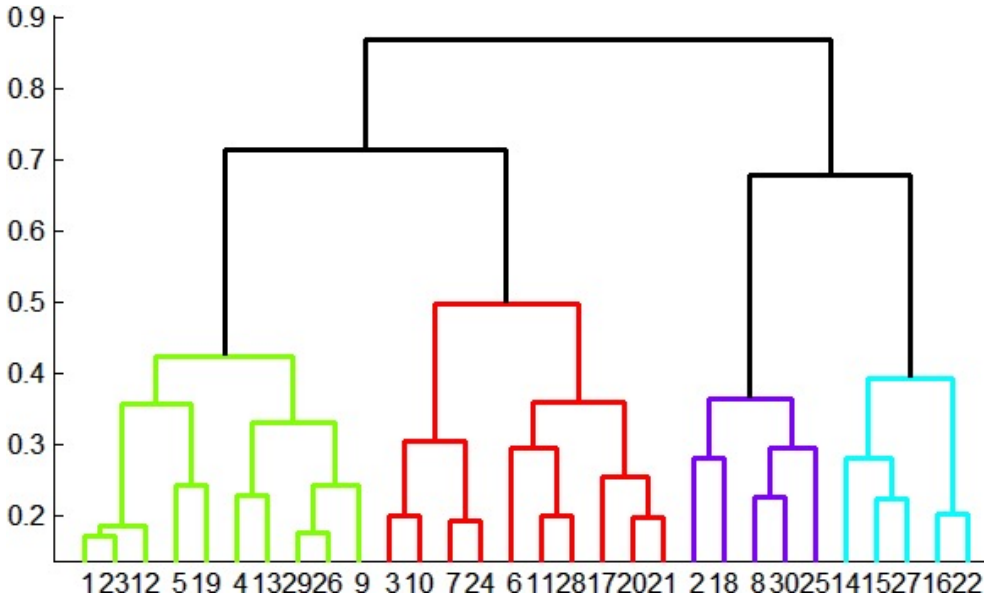
Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

STAT/BIOST 527

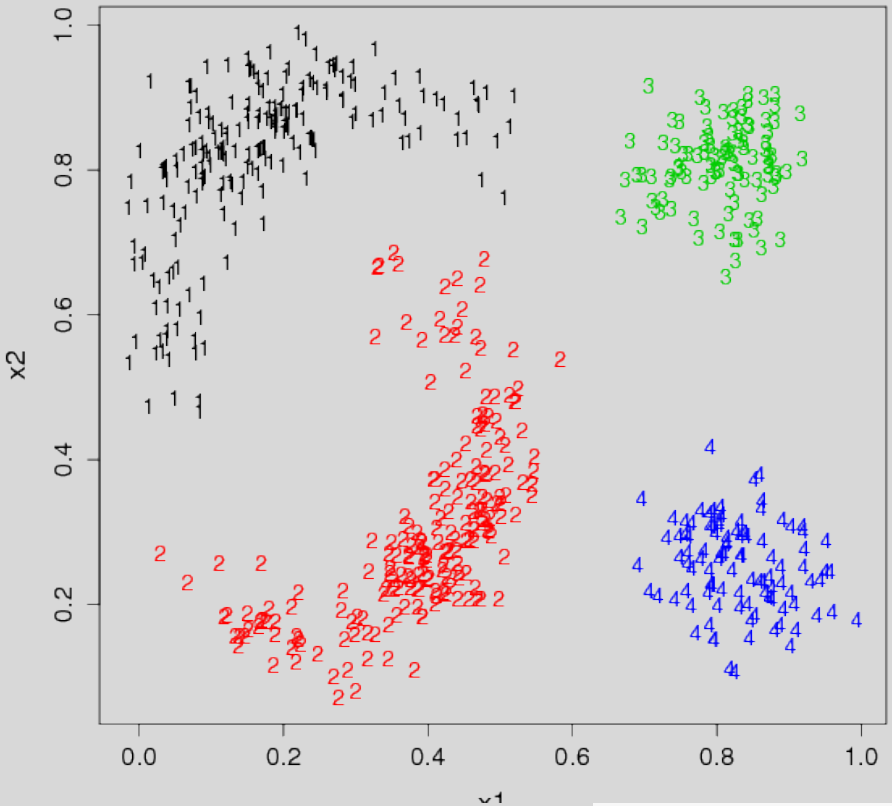
Hierarchical Methods of Clustering

- **Agglomerative** (bottom up):
  - Initially, each point is a cluster
  - Repeatedly combine the two "nearest" clusters into one
- **Divisive** (top down):
  - Start with one cluster and recursively split it

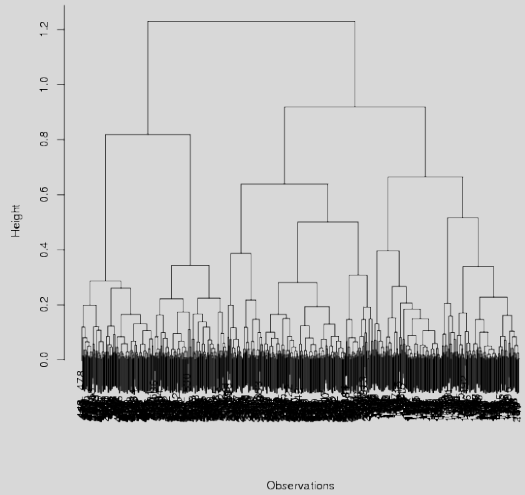


What is hierarchical clustering?

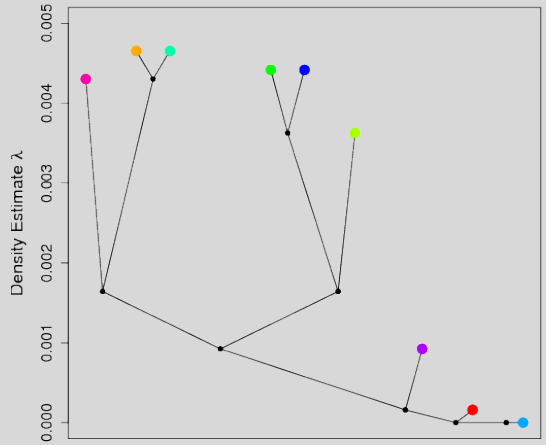
- Clusters have cluster structure
- Represented by
  - **Dendrogram**
  - **Cluster Tree**
    (only from KDE)



Dendrogram



Cluster Tree



Hierarchical clustering – Overview

(Dendrograms)

- **Agglomerative** (bottom up)
  - **Single linkage**
    - based on Minimum Spanning Tree
    - $\mathcal{O}(n^2 \log n)$
    - sensitive to outliers
  - Heuristics – average linkage
  - **Agglomerative K-means**
    - Loss $\mathcal{L}(\Delta_K) = 0$ for $K = n$
    - When $K \leftarrow K - 1$ (two clusters merged), $\mathcal{L}$ increases
    - For $K = n, n-1, \ldots 2$, iteratively merge the 2 clusters that minimize increase of $\mathcal{L}$
    - $\mathcal{O}(n^3)$ – too expensive for big data

- **Divisive** (bottom down)
  - Recursively split $\mathcal{D}$ into $K = 2$ clusters
  - almost any clustering algorithm (e.g. K-means, min diameter)
  - notable example Spectral clustering (later)

  - Advantages
    - most important splits are first
    - can stop after only a few splits

Single Linkage Clustering

Algorithm Single-Linkage

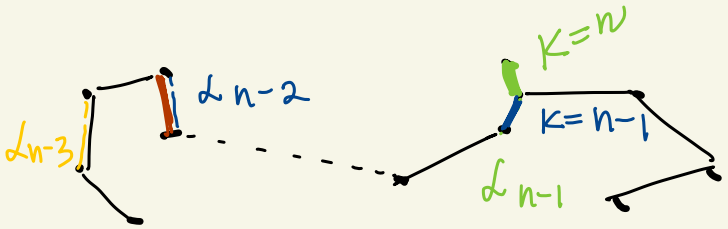
Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, number clusters K

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!

Loss $\mathcal{L}(\Delta_K)$ = at level K $\quad K = 1:n$

1. Single Linkage $\alpha_k \equiv \mathcal{L}(\Delta_k) = \min_{k \neq k'} \min_{\substack{i \in C_k \\ j \in C_{k'}}} \|x^i - x^j\|$

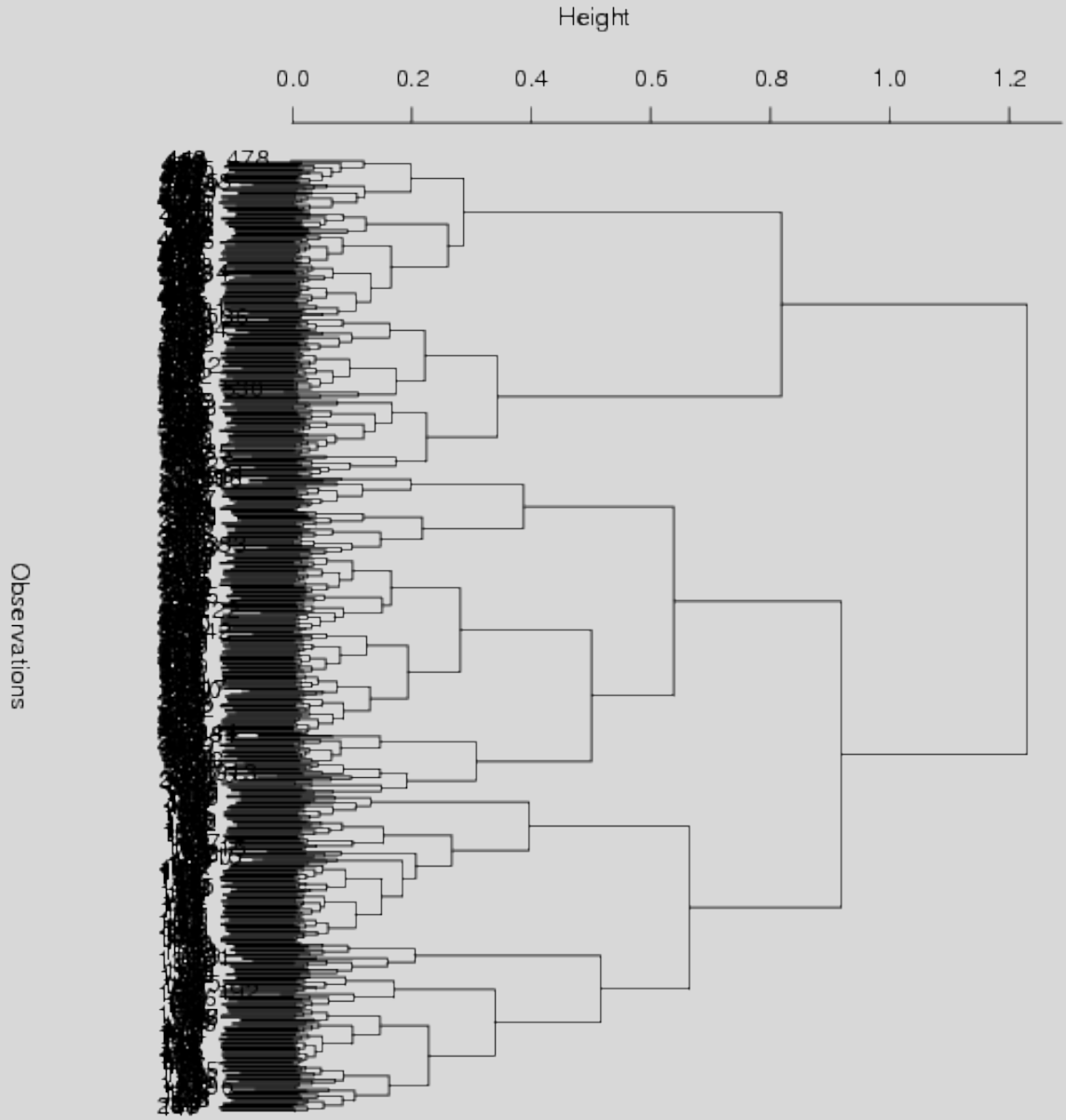


Agglomeration

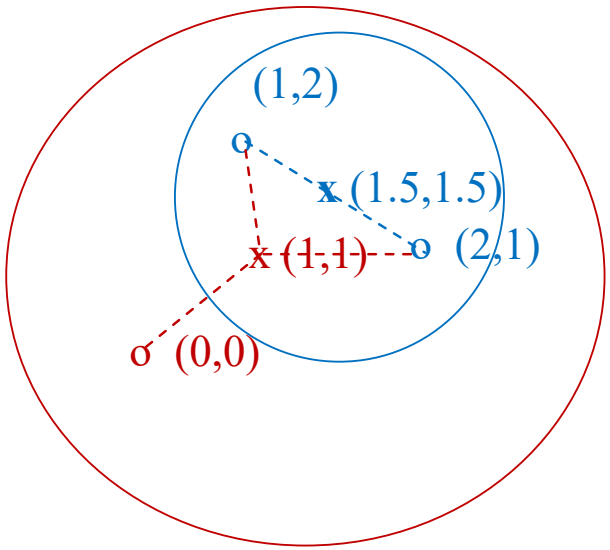
$K = n$
$K = n-1$
$K = n-2$
...

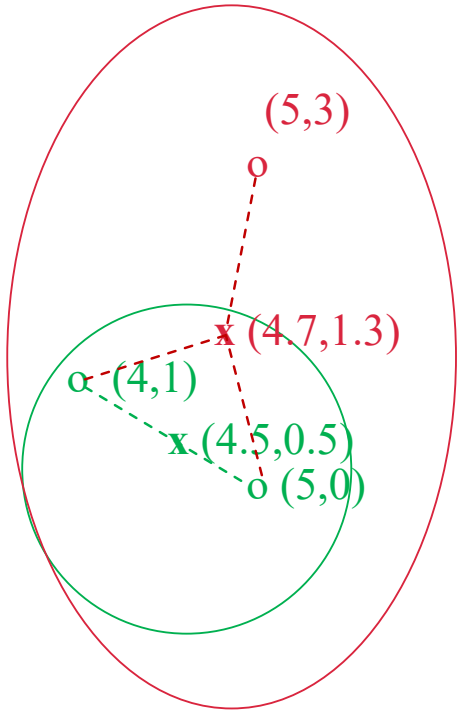
$\Rightarrow$ tree (no cycles)
(MST)
Min Spanning Tree
$\Downarrow$
can be run as Divisive too !!

Pb: NOT ROBUST


Height
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Observations

Example: Hierarchical clustering

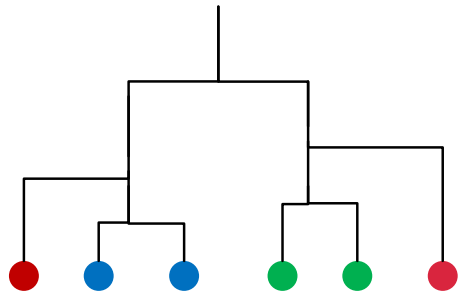




Data:

o … data point

x … centroid



Dendrogram