CSE8803/CX4803 Machine Learning in Computational Biology

Lecture 9: Learning from high-dimensional data:
UMAP, clustering

Xiuwei Zhang

School of Computational Science and Engineering

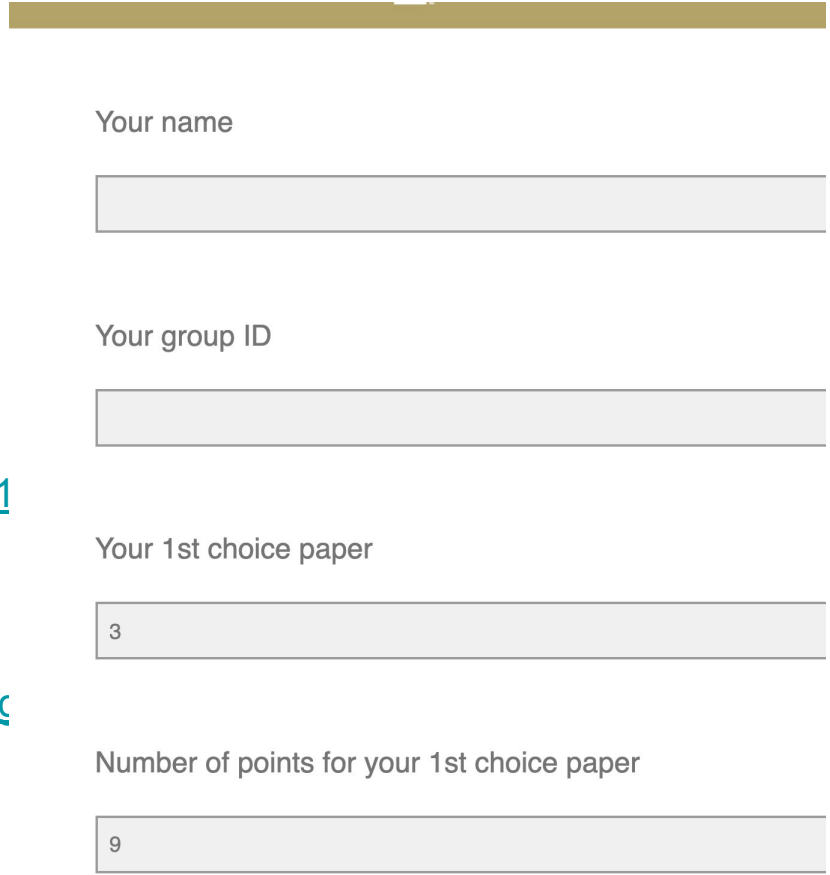
Paper presentation logistics

Presentation dates are fixed for all groups

Now start paper bidding for Phase 1 presentations! Deadline Thursday 11:59pm.

Bidding link: https://gatech.co1.qualtrics.com/jfe/form/SV_0VbI3SXv1pCTlk

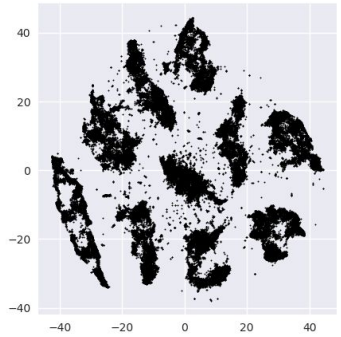
List of papers: https://docs.google.com/document/d/1RJDWddTV3hqnc6YGzSXGpKCIYgjL5k_GtLarOU54s/edit



t-SNE

Why its output (the low-dimensional representations of data) is more often used for visualization than for further downstream analysis (clustering, etc)?

- It doesn’t learn a function from high-dim data to low-dim, so if new data points come we can’t directly convert it to low-dim.
- It doesn’t directly preserve distance but rather preserves the neighborhood of every datapoint. So the distance between points in low-dim space can’t be interpreted as close representation of the original distance. Distance-based clustering methods should be used with caution to the output of tSNE. In particular, larger distances are not preserved.
- It’s output is often good for human eyes (also considering the effect of varying parameters), but not good for automatic clustering methods like k-means.
- Still controversial.



UMAP Uniform Manifold Approximation and Projection

MDS: preserving distance

tSNE: preserving neighborhood

UMAP: preserving graph topology

Compared to t-SNE, UMAP seems to be

- faster
- deterministic

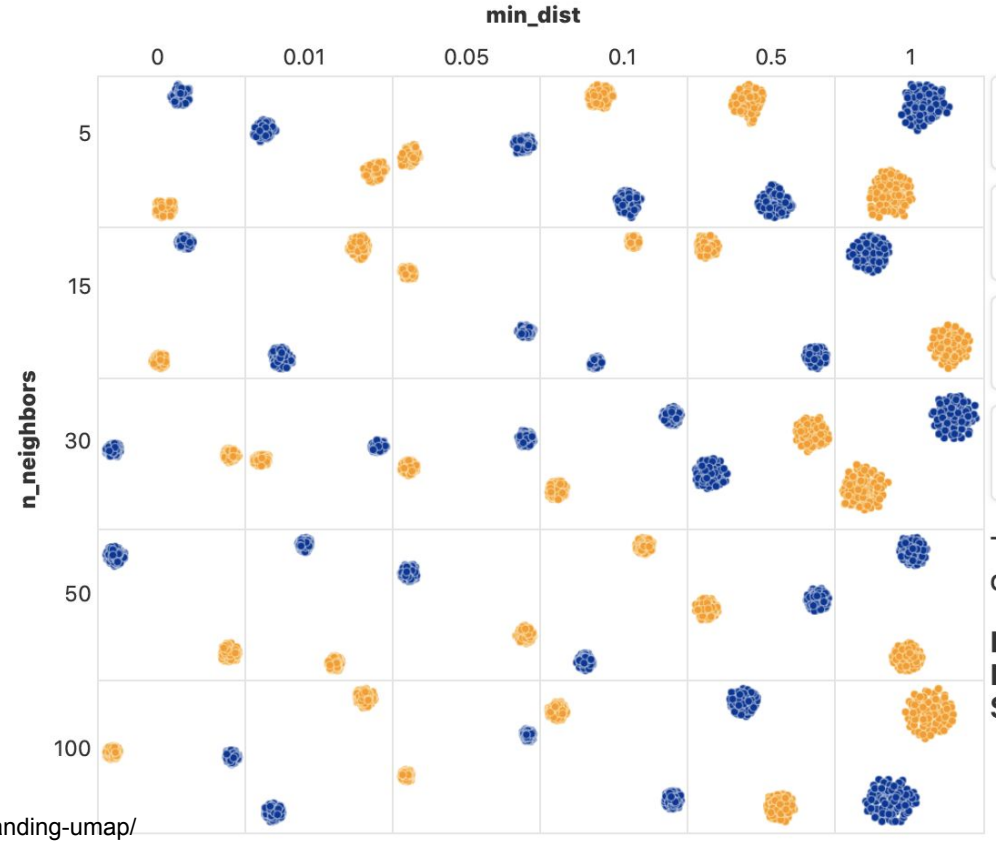
Very similar intuition but different mathematical framework

UMAP Uniform Manifold Approximation and Projection

Hyper-parameters

N_neighbors: the number of nearest neighbors to consider

Min_dist: minimum distance apart that points are allowed to be in the low dimensional representation



Two clusters with equal numbers of points, but different variances within the clusters.

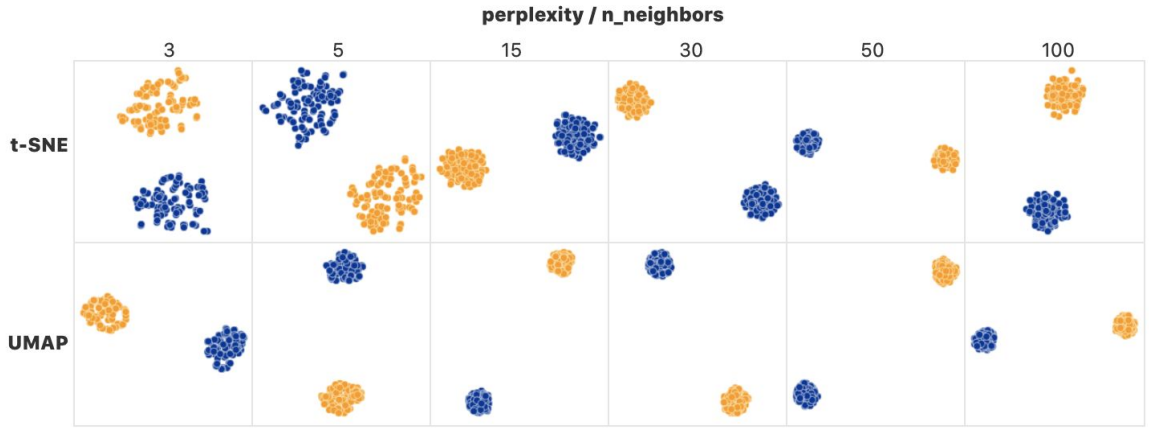
https://pair-code.github.io/understanding-umap/

UMAP Uniform Manifold Approximation and Projection

Hyper-parameters

N_neighbors: the number of nearest neighbors to consider

Min_dist: minimum distance apart that points are allowed to be in the low dimensional representation



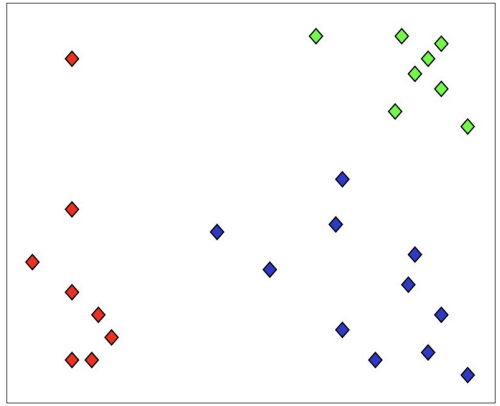
Two clusters with equal numbers of points.

https://pair-code.github.io/understanding-umap/

Clustering algorithms

Clustering - unsupervised learning methods

- Organizing data into clusters such that there is
  - high intra-cluster similarity
  - low inter-cluster similarity
- Informally, finding natural groupings among objects.
- Applications
  - Organizing data into clusters provides information about the internal structure of the data
  - Sometimes the partitioning is the goal
  - Knowledge discovery in data



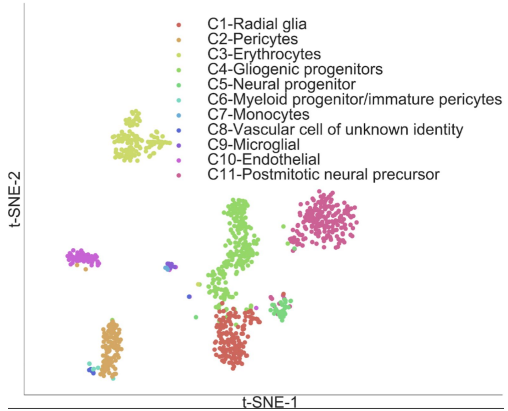
K=2

Original

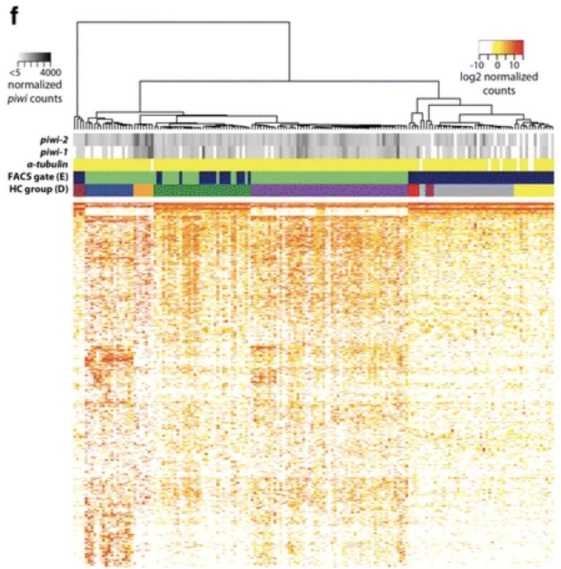
Clustering algorithms

Clustering - unsupervised learning methods

- K-means (and fuzzy k-means)
- Hierarchical clustering
- Spectral clustering
- Graph-based : Louvain, Leiden clustering

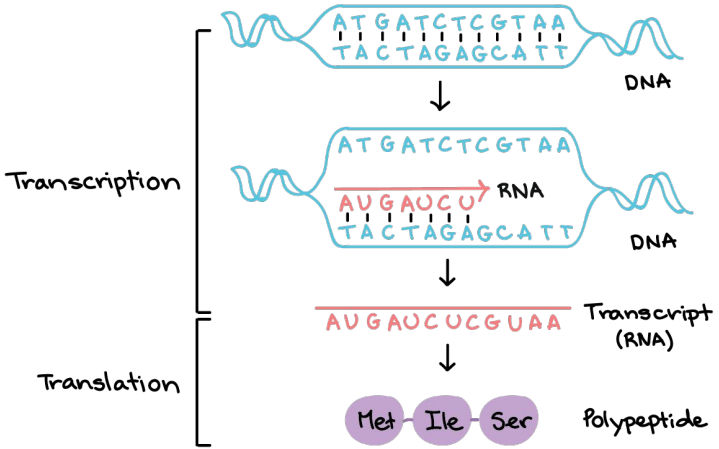


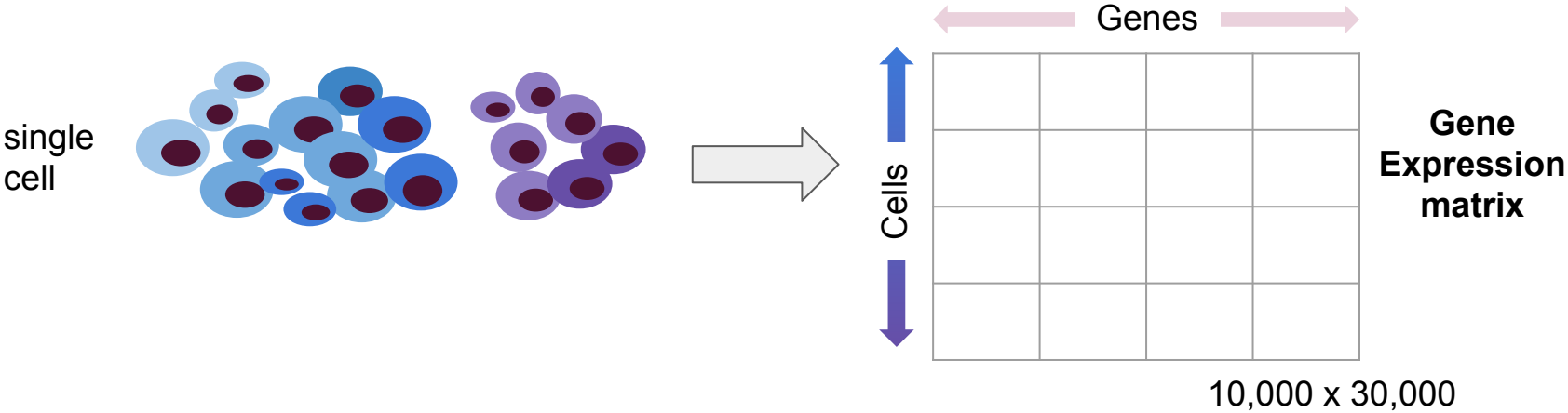
https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007040



https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0937-9/figures/1

Some biological background: Gene expression analysis





k-means

Given the number of clusters *k*

An iterative clustering algorithm.

Initialize:

Pick k random points as cluster centers

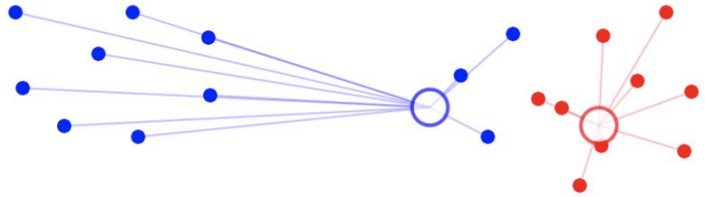
Alternate:

1. Assign data points to closest cluster center
2. Change the cluster centers to the average of its assigned points

Stop:

When no points' assignments change

k-means

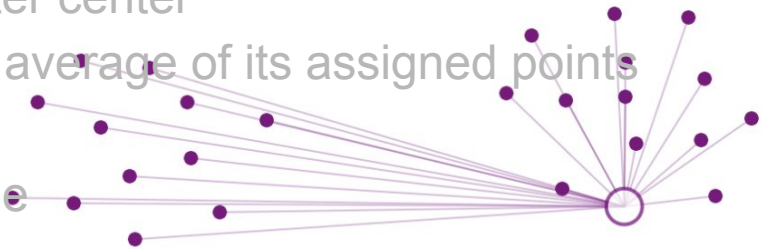


Initial locations of centers

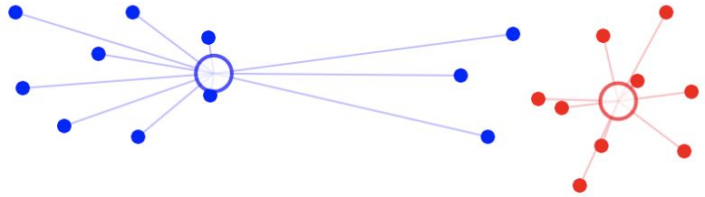
Points are assigned to closest centers

Given the number of clusters *k*
An iterative clustering algorithm.

Initialize:
Pick k random points as cluster centers
Alternate:
1. Assign data points to closest cluster center
2. Change the cluster centers to the average of its assigned points
Stop:
When no points' assignments change



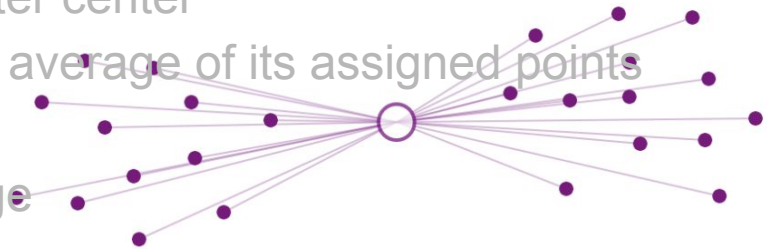
k-means



Given the number of clusters *k*
An iterative clustering algorithm.

Centers updated as the average of all points of that label

Initialize:
Pick k random points as cluster centers
Alternate:
1. Assign data points to closest cluster center
2. Change the cluster centers to the average of its assigned points
Stop:
When no points' assignments change

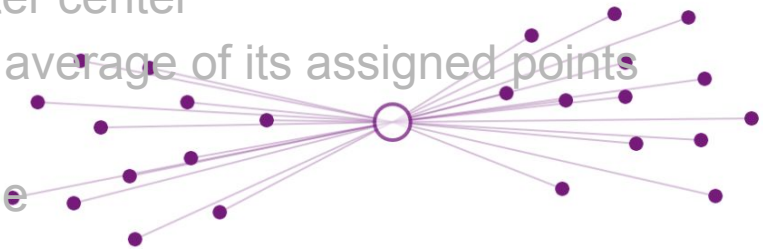


k-means

Given the number of clusters *k*
An iterative clustering algorithm.

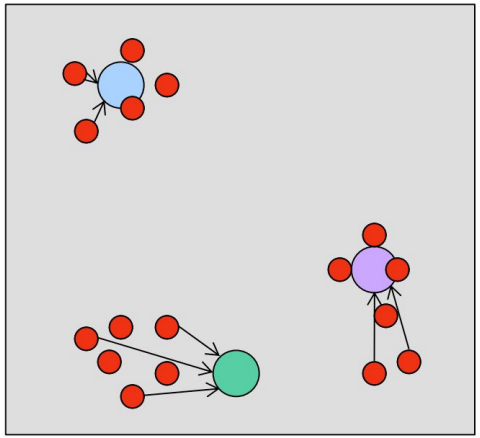
Labels for each point updated

Initialize:
Pick k random points as cluster centers
Alternate:
1. Assign data points to closest cluster center
2. Change the cluster centers to the average of its assigned points
Stop:
When no points' assignments change



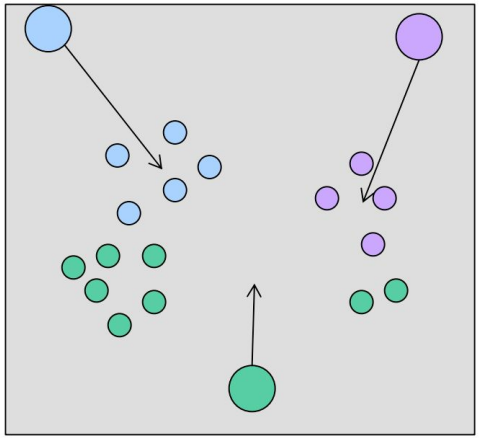
https://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means/

K-means as an “expectation maximization” process



(“M”)

(“E”)



Re-assign each point $\mathbf{x}_i$ to nearest center k

➔ Minimize distance from $\mathbf{x}_i$ to $\boldsymbol{\mu}_k$:

$$d_{i,k} = \left(\mathbf{x}_i - \boldsymbol{\mu}_k\right)^2$$

Update center $\boldsymbol{\mu}_k$ to the mean of the points assigned to it:

$$\boldsymbol{\mu}_k = \sum_{\mathrm{x_i\ with\ label\ j}} \frac{\mathbf{x}_i}{\left|\mathbf{x}^k\right|}$$

where: $\left|\mathbf{x}^k\right| = \#\mathbf{x}_i$ with label k $= n_k$

Slide credit: Manolis Kellis

K-means as an “expectation maximization” process

$$\mathrm{COST}\left(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,...,\mathbf{x}_n\right)=\sum_{\boldsymbol{\mu}_k}\sum_{\mathbf{x}_i \text{ with label k}}\left(\mathbf{x}_i-\boldsymbol{\mu}_k\right)^2$$

The M step:
With the current the cluster assignment (labels) of points

$$\boldsymbol{\Sigma}_{\text{xi with label k}}(\mathbf{x}_i-\boldsymbol{\mu}_k)^2=\boldsymbol{\Sigma}_{\text{xi with label k}}(\mathbf{x}_i^2-2\mathbf{x}_i\boldsymbol{\mu}_k+\boldsymbol{\mu}_k^2)$$

$$=\boldsymbol{\Sigma}_{\text{xi with label k}}\mathbf{x}_i^2-2\boldsymbol{\mu}_k\boldsymbol{\Sigma}_{\text{xi with label k}}\mathbf{x}_i+n_k\boldsymbol{\mu}_k^2$$

Take derivative on $\boldsymbol{\mu}$:

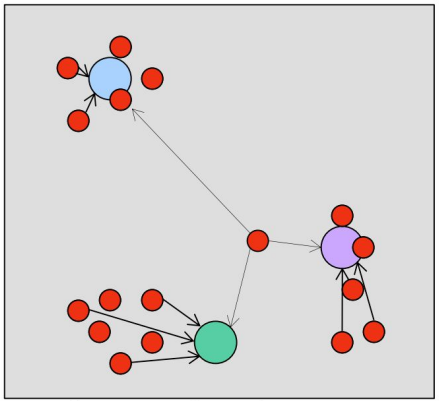
$$2\boldsymbol{\Sigma}_{\text{xi with label k}}\mathbf{x}_i=2n_k\boldsymbol{\mu}_k$$

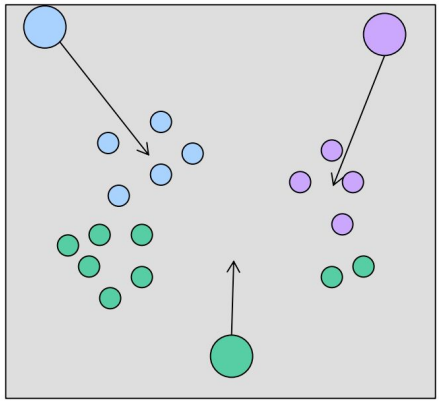
$$\boldsymbol{\mu}_k=1/n_k\,\boldsymbol{\Sigma}_{\text{xi with label k}}\mathbf{x}_i$$

Fuzzy k-means

What if some points are half-way between two cluster centers?

Assign partial weights.





Re-assign each point $\mathbf{x}_i$ to **all** centers, **weighted by distance**

➔ For each point calculate the probability of membership for each category K:

$$P(\text{label K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k)$$

Update center $\boldsymbol{\mu}_k$ to the **weighted mean** of the points assigned to it:

$$\boldsymbol{\mu}_k = \sum_{\mathbf{x}_i \text{ with label j}} \mathbf{x}_i \, P(\boldsymbol{\mu}_k \mid \mathbf{x}_i)^b \Big/ \sum_{\mathbf{x}_i \text{ with label j}} P(\boldsymbol{\mu}_k \mid \mathbf{x}_i)^b$$

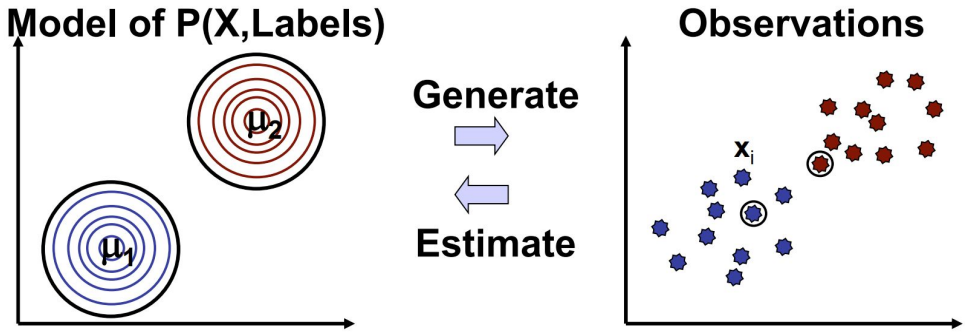
Regular K-Means is a special case of fuzzy k-means where:

$$P(\text{label K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k) = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is closest to } \boldsymbol{\mu}_k \\ 0 & \text{otherwise} \end{cases}$$

19

Slide credit: Manolis Kellis

K-means as a generative model



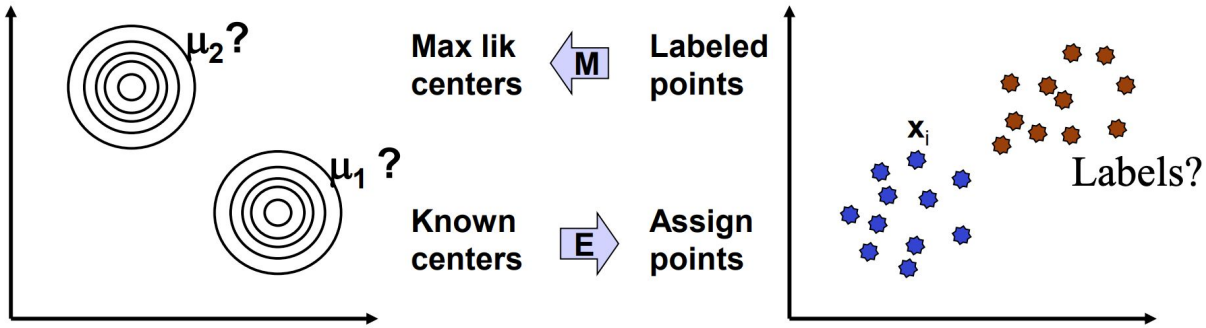
Consider that each point is generated from a Gaussian distribution with unit variance.

$$P(x_i|\mu_k) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x_i - \mu_k)^2}{2}\right\}$$

Problem: find cluster centers and labels for data points such that the total likelihood is maximized

Solution is the same as the k-means solution! (equivalence)

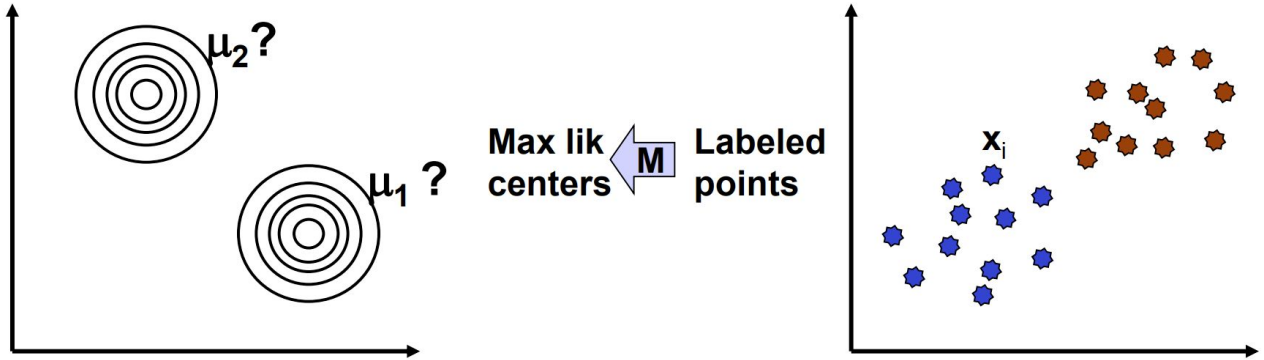
K-means as a generative model



Problem: find cluster centers and labels for data points such that the total likelihood is maximized

E step: If centers are known → Estimate memberships
M step: If assignments known → Compute centroids

K-means as a generative model



$$\arg\max_{\mu} \left\{ \log \prod_i P\left(x_i \mid \mu\right) \right\} = \arg\max_{\mu} \sum_i \left\{ -\frac{1}{2}\left(x_i - \mu\right)^2 + \log\left(\frac{1}{\sqrt{2\pi}}\right) \right\}$$

Seeking the max likelihood estimate of the cluster mean

$$= \arg\min_{\mu} \sum_i \left(x_i - u\right)^2$$

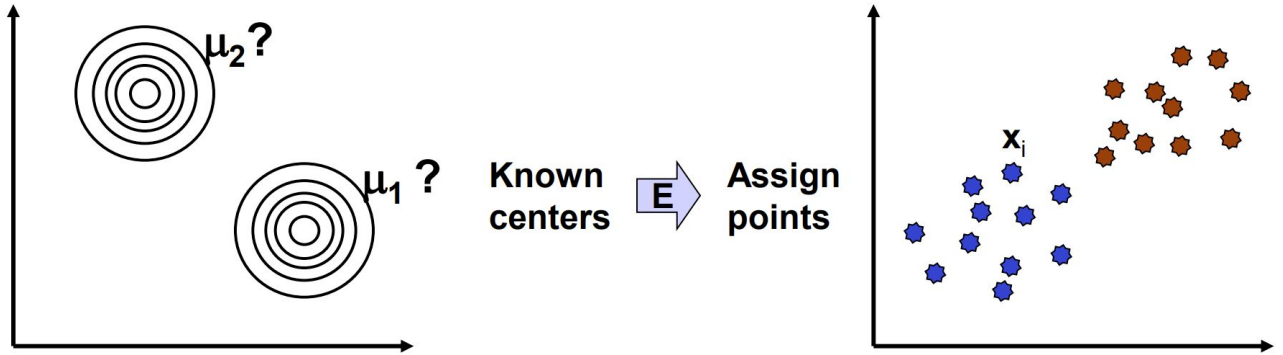
Solution is the centroid of the x_i

Equivalent

EM solution ⟺ **K-means solution**

Slide credit: Manolis Kellis

K-means as a generative model



$$\arg\max_k P_k(\mathbf{x}_i \mid \boldsymbol{\mu}_i) = \arg\max_k \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{u}_k)^2}{2}\right\} = \arg\min_k (\mathbf{x}_i - \mathbf{u}_k)^2$$

Seeking the label k that maximizes likelihood of point

Solution is the nearest center

EM solution ⟺ **K-means solution** (Equivalent)

Slide credit: Manolis Kellis

Comparison between methods

	K-means		Fuzzy K-means	
	algorithmic formulation	probabilistic interpretation	algorithmic formulation	probabilistic interpretation
Initialization	Initialize K centers $\boldsymbol{\mu}_k$	Initialize model parameters	Initialize K centers $\boldsymbol{\mu}_k$	Initialize model parameters
E-step: Estimate prob of hidden labels (point assignments to classes)	Assign $\mathbf{x}_i$ label of nearest center distance $d_{i,k} = (\mathbf{x}_i - \boldsymbol{\mu}_k)^2$	Estimate most likely missing label given previous parameters	Calculate probability of membership for each point to each class $P(\text{label } K \mid \mathbf{x}_i, \boldsymbol{\mu}_k)$	Estimate probability over missing labels given previous parameters
M-step: Update params to max likelihood estimates given assignments	Move $\boldsymbol{\mu}_k$ to centroid of all points with that label	Choose new max likelihood params given points in label	Move $\boldsymbol{\mu}_k$ to weighted centroid of all points, each weighted by P(label)	Choose new params to maximize expected likelihood given label estimates
Iteration	Iterate	Iterate	Iterate	Iterate

Slide credit: Manolis Kellis

Characteristics of k-means

K-means partitions the points into Voronoi cells

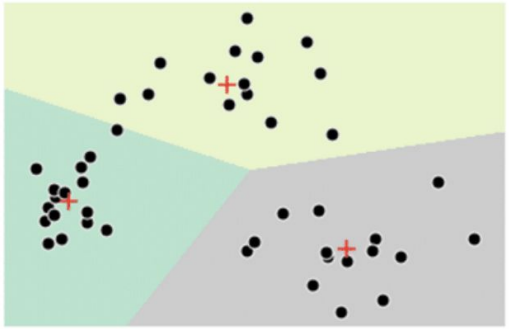
Does it converge?

Yes.

Does it guarantee optimality?

No. (Initialization matters)

Prior dimension reduction helpful



Disadvantages:

- Needs k as input
- Unable to handle noisy data and outliers
- Not suitable to discover clusters with non-convex shapes

Hierarchical clustering

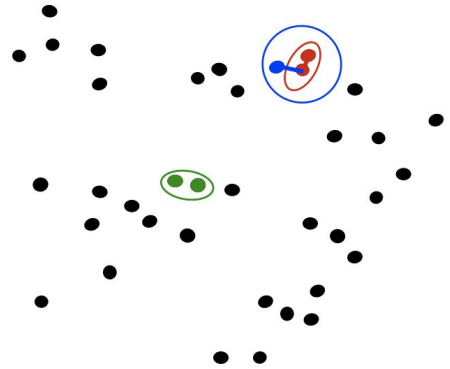
Idea: agglomerative clustering

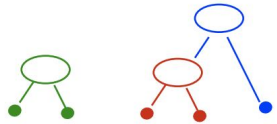
- First merge very similar instances
- Incrementally build larger clusters out of smaller clusters

Algorithm: Maintain a set of clusters

- Initially, each instance in its own cluster
- Repeat:
  - Pick the two closest clusters
  - Merge them into a new cluster
  - Stop when there's only one cluster left

How to define?





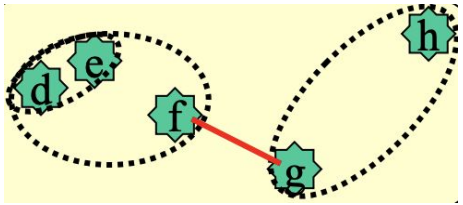
Output:

Produces not one clustering, but a family of clusterings represented by a dendrogram

Hierarchical clustering: define “closest” for clusters

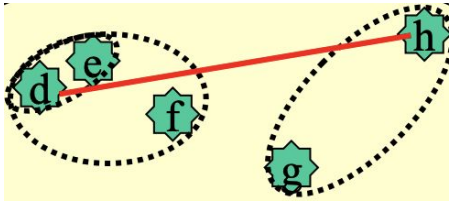
Closest pair (single-link)

$$CD(X,Y)=\min_{x \in X,\, y \in Y} D(x,y)$$



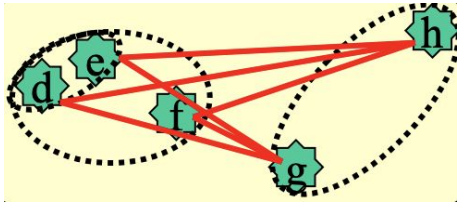
Furthest pair (complete-link)

$$CD(X,Y)=\max_{x \in X,\, y \in Y} D(x,y)$$



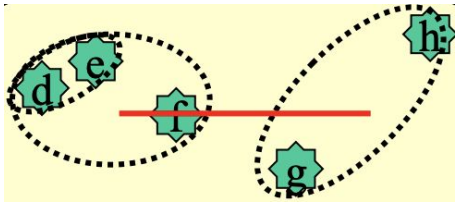
average-link

$$CD(X,Y)=\text{avg}_{x \in X,\, y \in Y} D(x,y)$$



centroids method

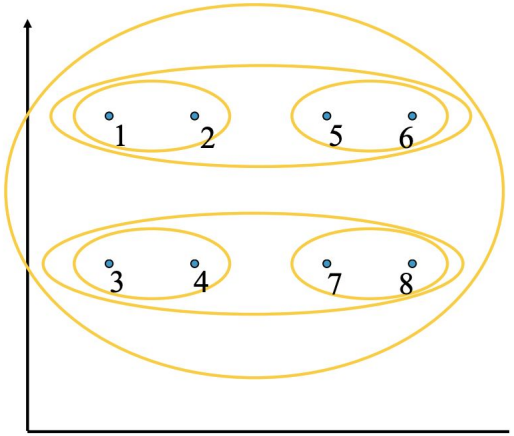
$$CD(X,Y)=D(\text{avg}(X), \text{avg}(Y))$$



Hierarchical clustering: define “closest” for clusters

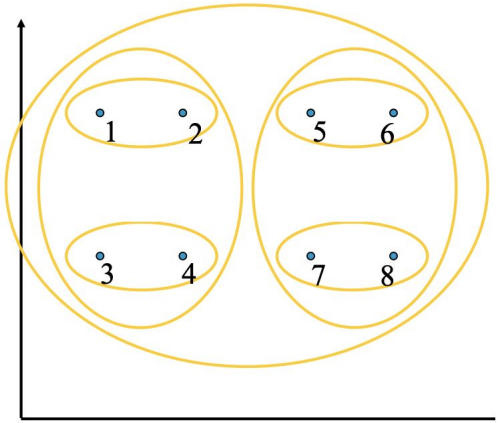
Cluster distance affects both results and runtime

Closest pair
(single-link clustering)



Potentially long and skinny clusters

Farthest pair
(complete-link clustering)

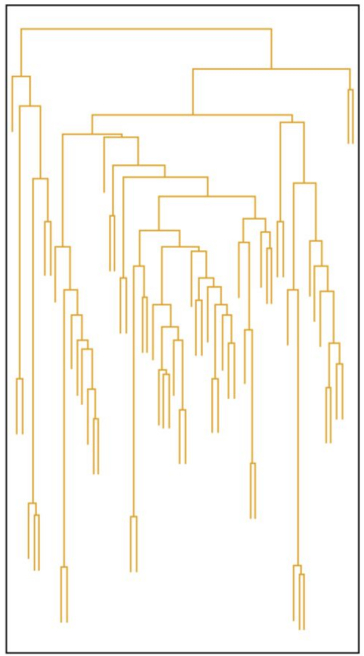


tight clusters

Figure credit: Thorsten Joachims

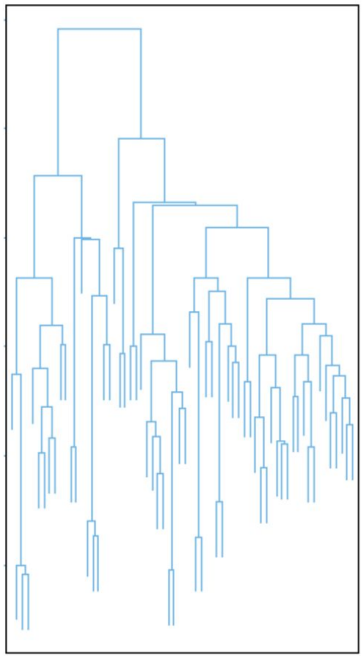
Hierarchical clustering: define “closest” for clusters

Average

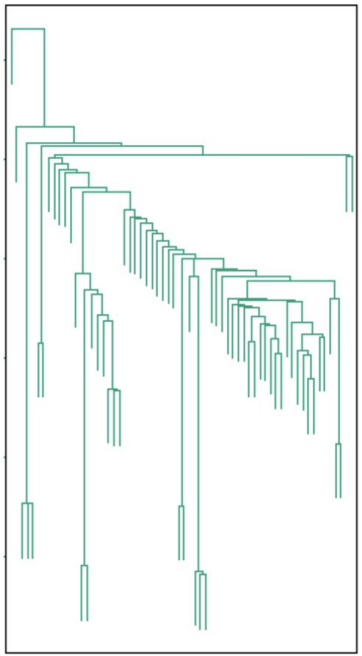


Robust to noise
Widely used

Farthest



Nearest



Centroid

fast

Figure credit: David Sontag

Hierarchical clustering: distance measures

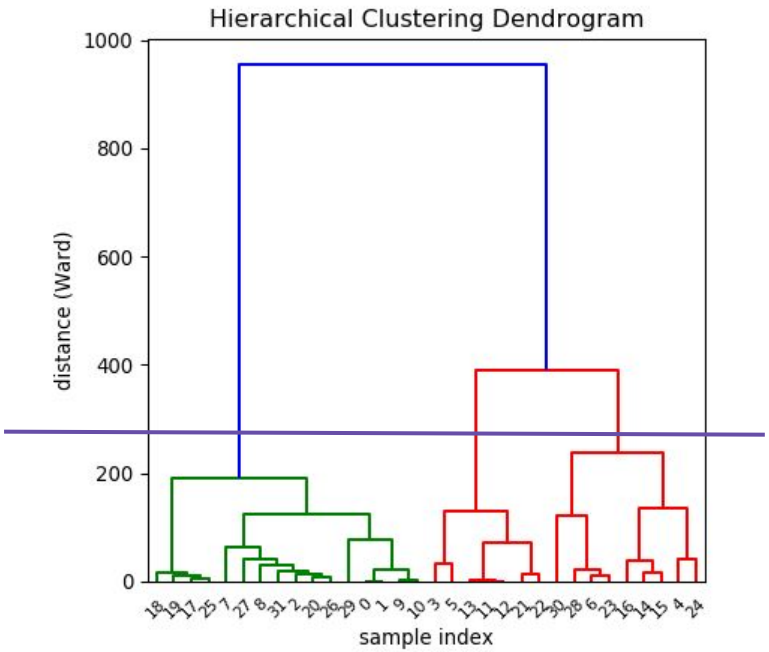
Table 1 Gene expression similarity measures

Manhattan distance (city-block distance, L1 norm)	$d_{fg} = \sum_c \left\| e_{fc} - e_{gc} \right\|$
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (\mathrm{e}_f - \mathrm{e}_g)' \Sigma^{-1} (\mathrm{e}_f - \mathrm{e}_g)$, where Σ is the (full or within-cluster) covariance matrix of the data
Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (angular separation, cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2 \sum_c e_{gc}^2}}$
Spellman rank correlation	As Pearson correlation, but replace $\boldsymbol{e_{gc}}$ with the rank of $\boldsymbol{e_{gc}}$ within the expression values of gene g across all conditions $\boldsymbol{c = 1 \ldots C}$
Absolute or squared correlation	$d_{fg} = 1 - \left\| r_{fg} \right\|$ or $d_{fg} = 1 - r_{fg}^2$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

https://www.nature.com/articles/nbt1205-1499

Hierarchical clustering: obtaining the clusters



Hierarchical clustering: Summary

- No need to specify the number of clusters in advance.
- Hierarchical structure maps nicely onto human intuition for some domains
- They do not scale well: where n is the number of total objects.
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results can be subjective