Machine Learning Week 10 Lecture: Clustering

Debasis Ganguly
Debasis.Ganguly@glasgow.ac.uk

School of Computing Science
University of Glasgow

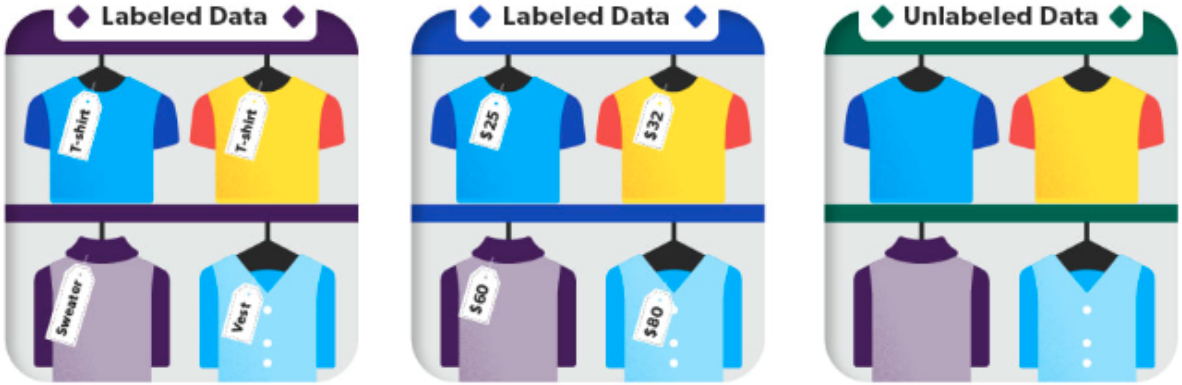
December 4, 2025

Aims

- This is the only lecture on clustering.
  - Understand what clustering is.
  - Understand the **K-means algorithm**.
  - Apply the idea of **kernels** to play around with the inter-similarity of items.
  - Understand the idea of **mixture models**.
  - Understand **Hierarchical Clustering Algorithms**.
  - Understand **evaluation of clustering**.

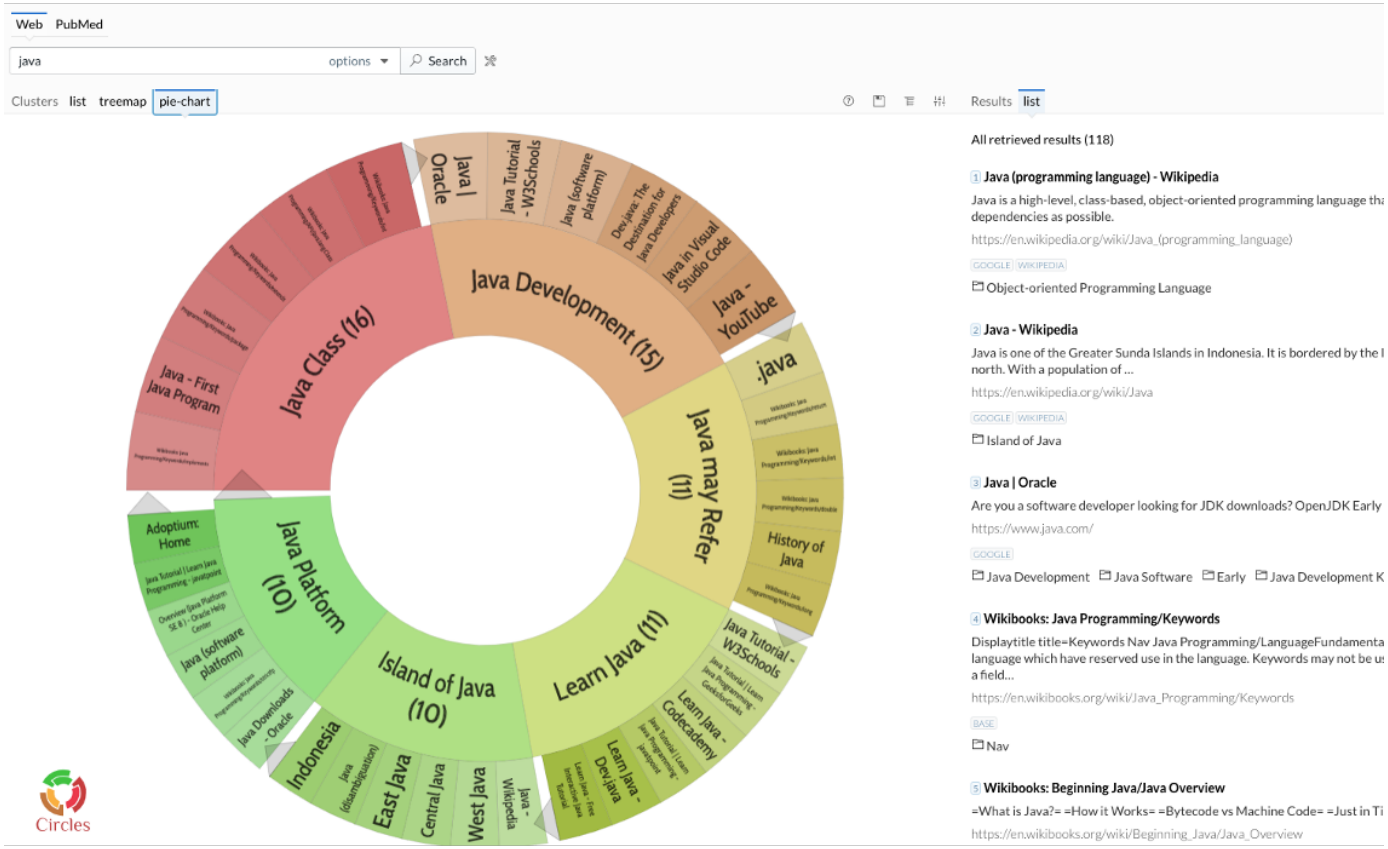
Introduction

Unsupervised learning



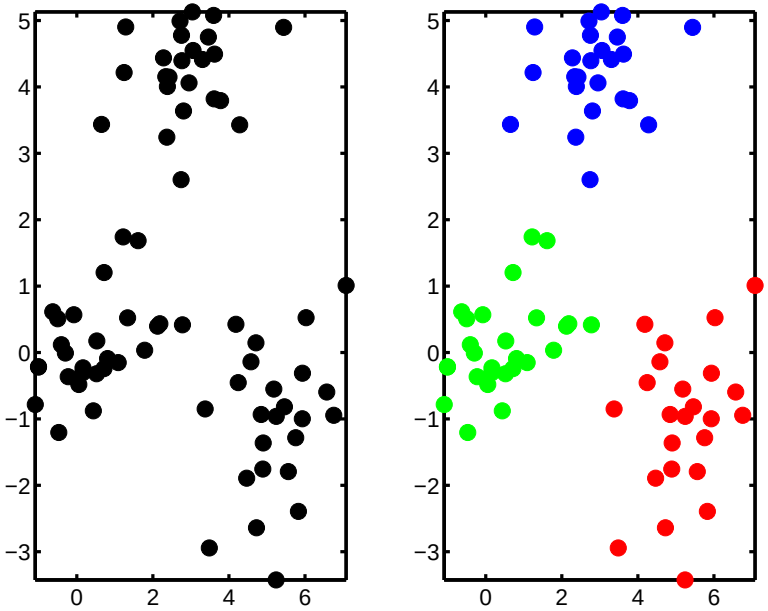
- Everything we've seen so far has been *supervised*.
  - We were given a set of $\mathbf{x}_n$ **and associated** t_n.
  - What if we just have $\mathbf{x}_n$?
- **Clustering**: Group items by various ways.
  - $\mathbf{x}_i$ indicating products a user $\mathbf{u}_j$ has bought.
  - Can group customers that buy similar products.
  - Can group products bought together.

Clustering Applications



- Helps to easily process large volumes of data by **grouping** them in terms of their **similarities**.
  - How to group?
  - How to define similarity?

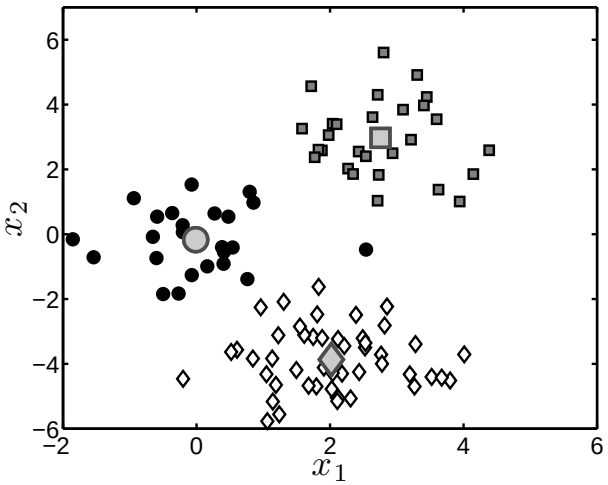
Clustering



- 2D example data:
- Data after clustering (points coloured according to cluster membership).
- Note that a clustering algorithm might have also *merged* the red and the green points.

K-means

Overall Idea of K-means



- What do we lack? Labels
- Assign them at random. This gives us **some hypothesis to work with**.
- Hypothesis itself can be bad because it was random!
- But with the hypothesis, we can now try to **explain the data**.

- Find cluster centroids.
- **Modify the initial hypothesis** by assigning **each point to its nearest centroid**.
- Assumption: A few bad choices cancel out with many good choices.

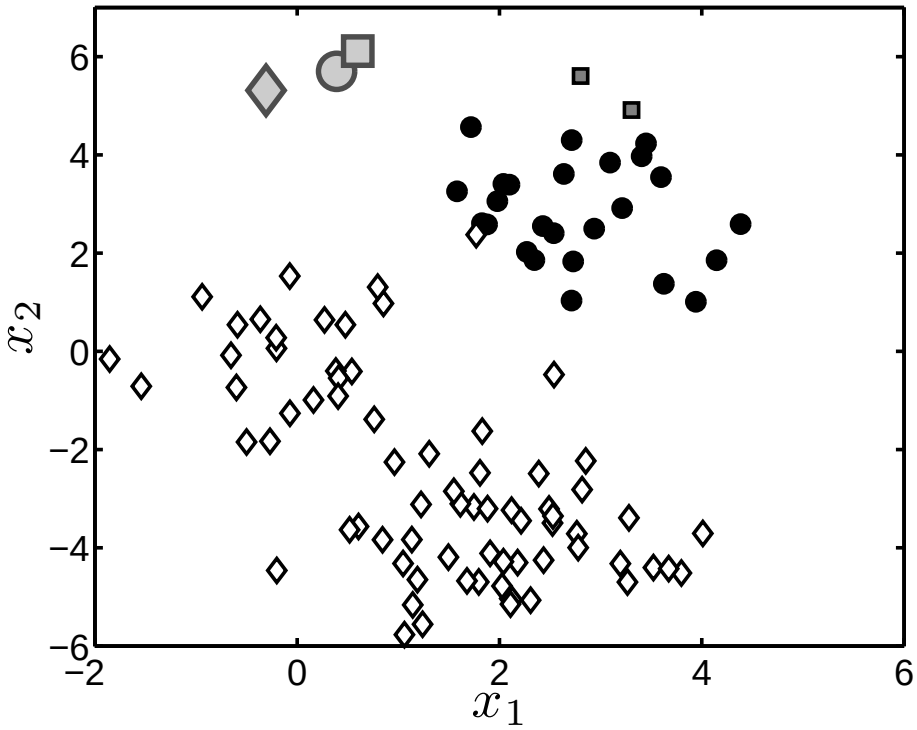
K-means Algorithm

- Guess $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
- Assign each $\mathbf{x}_n$ to its closest $\boldsymbol{\mu}_k$
- **Latent variables** for cluster membership: $z_{nk} \in \{0, 1\}$
- $z_{nk} = 1$ if $\mathbf{x}_n$ assigned to $\boldsymbol{\mu}_k$ (0 otherwise)
- Update $\boldsymbol{\mu}_k$ to average of $\mathbf{x}_n$s assigned to $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} z_{nk} \mathbf{x}_n}{\sum_{n=1}^{N} z_{nk}}$$

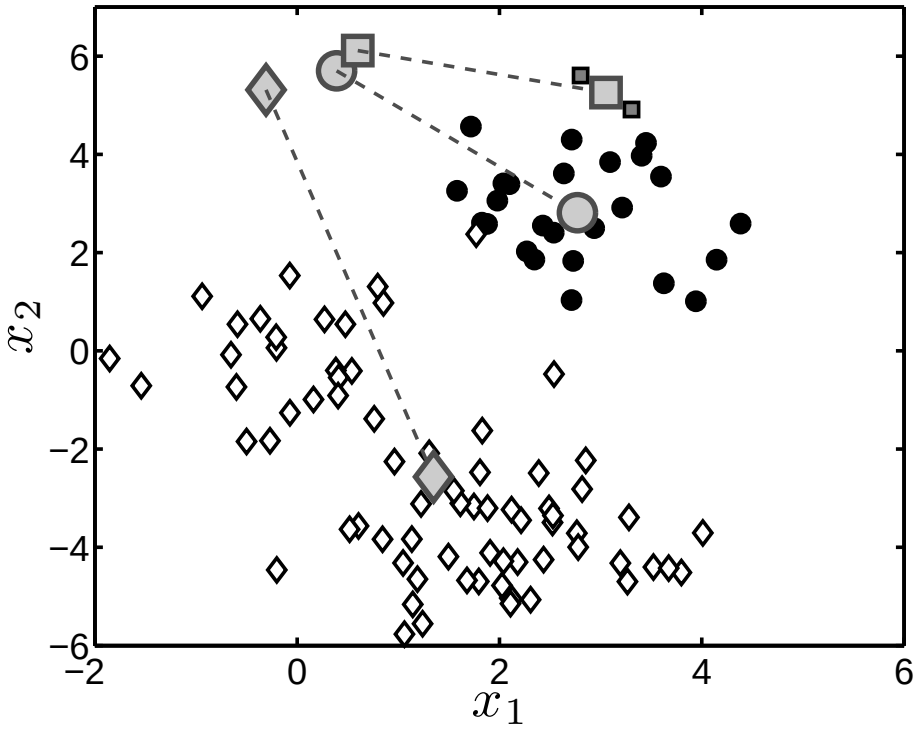
- Return to 2 until assignments do not change.
- K-means algorithm **eventually converges**. It will reach a point where the assignments don't change.

K-means – example



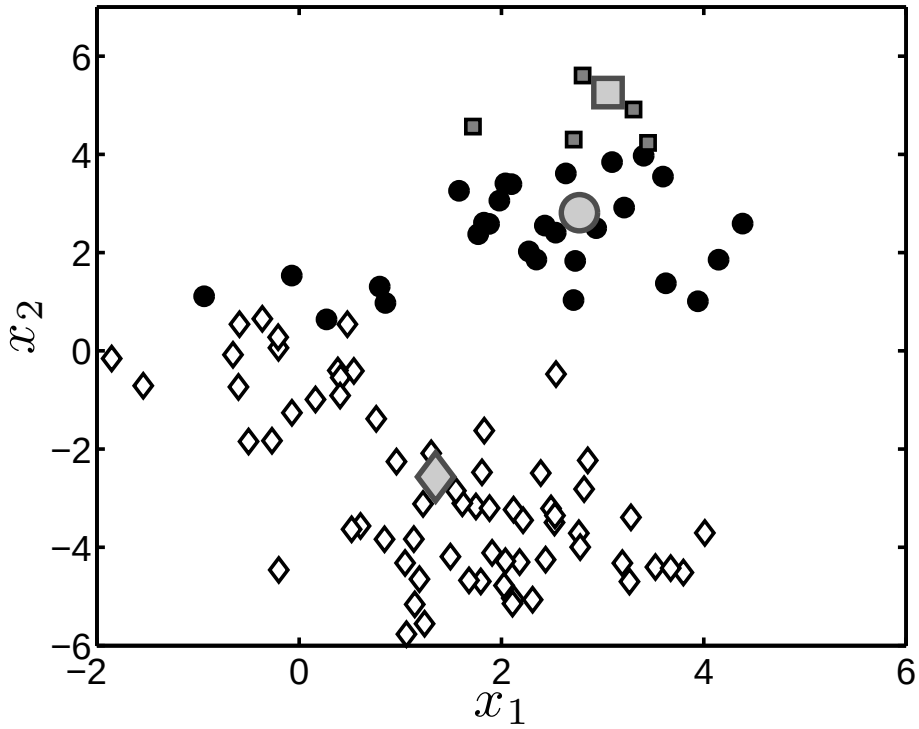
- Cluster means randomly assigned (top left).
- Points assigned to their closest mean.

K-means – example



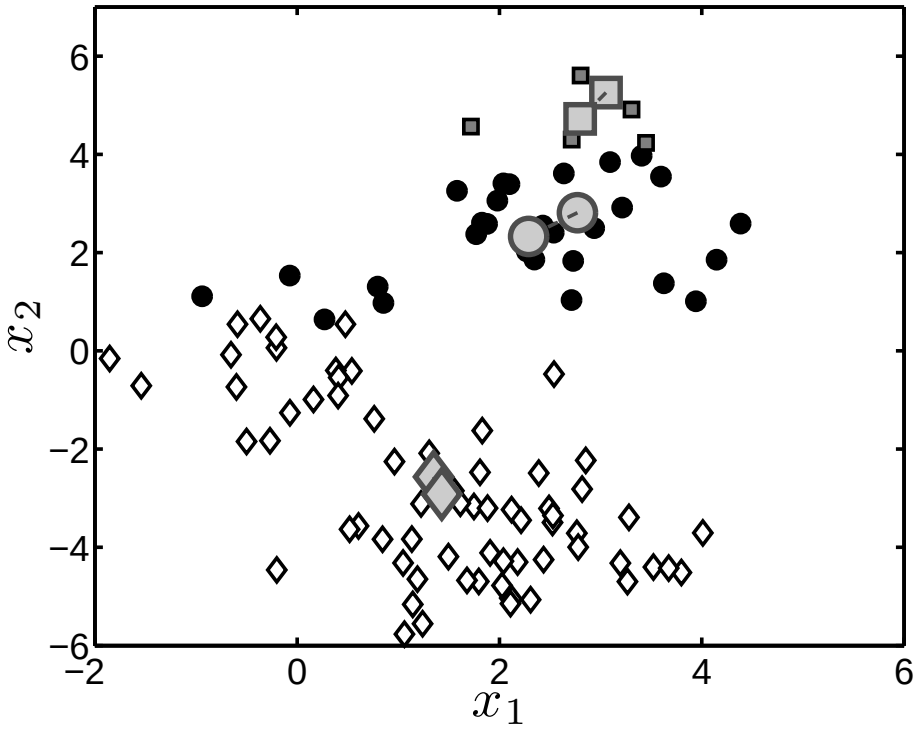
- Cluster means updated to mean of assigned points.

K-means – example



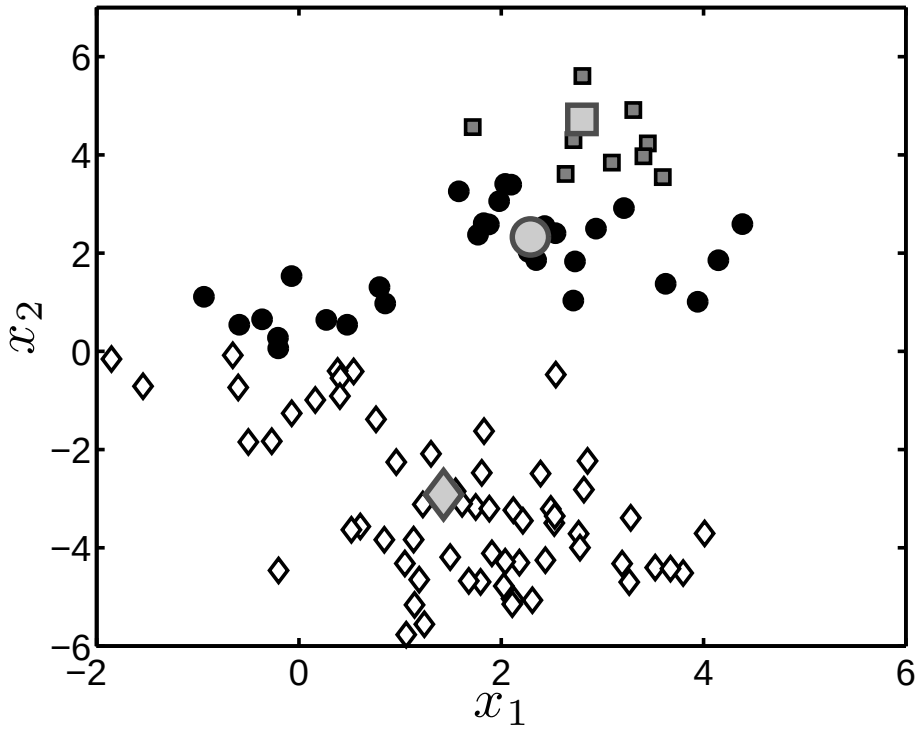
- Points re-assigned to closest mean.

K-means – example



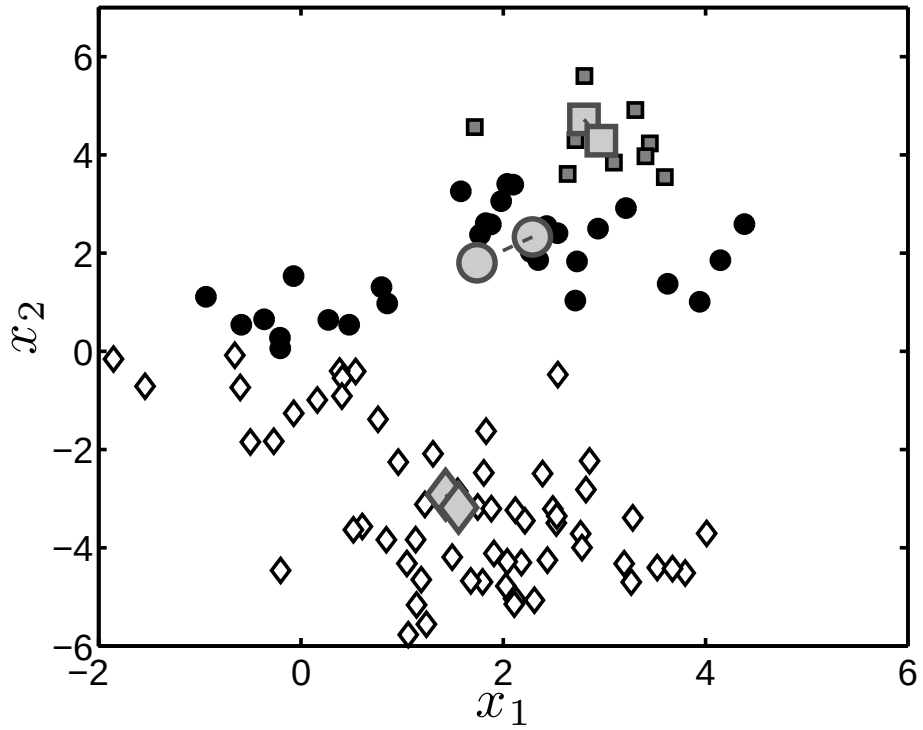
- Cluster means updated to mean of assigned points.

K-means – example



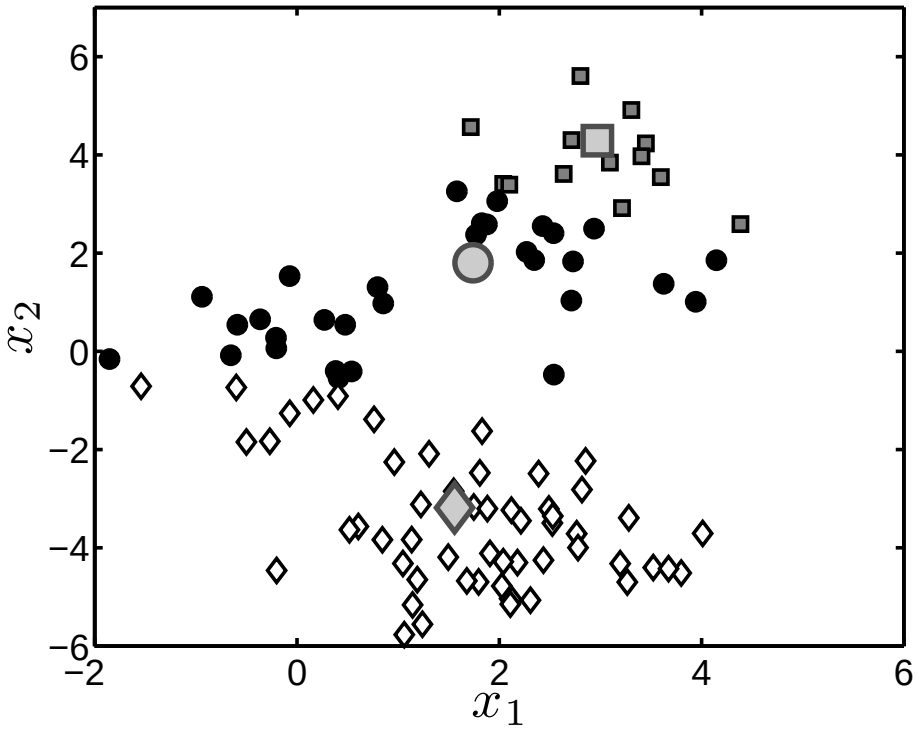
- Assign point to closest mean.

K-means – example



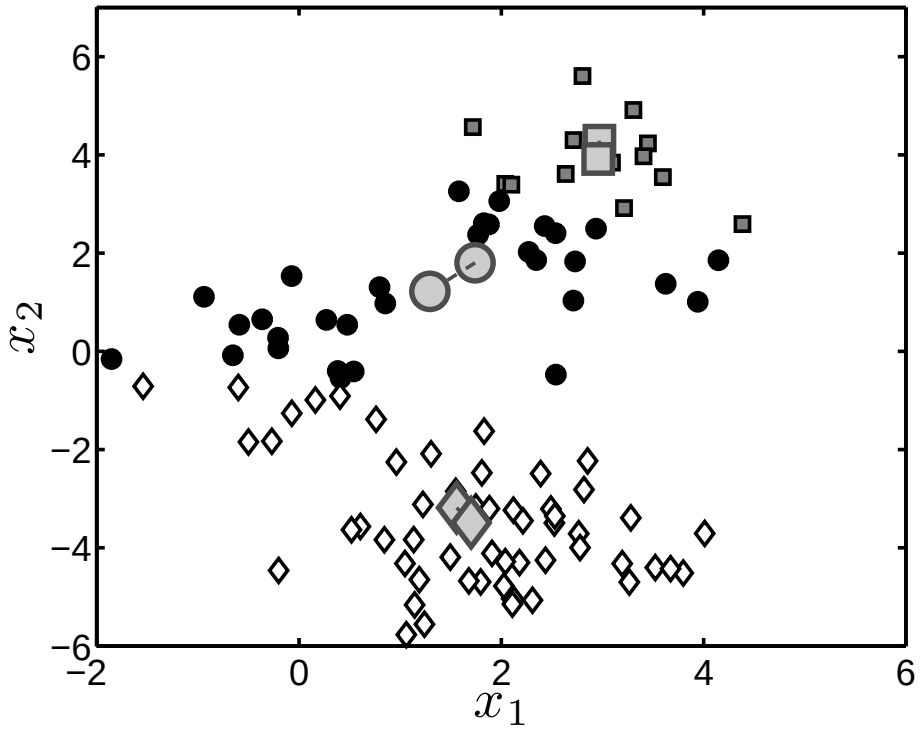
- Update mean.

K-means – example



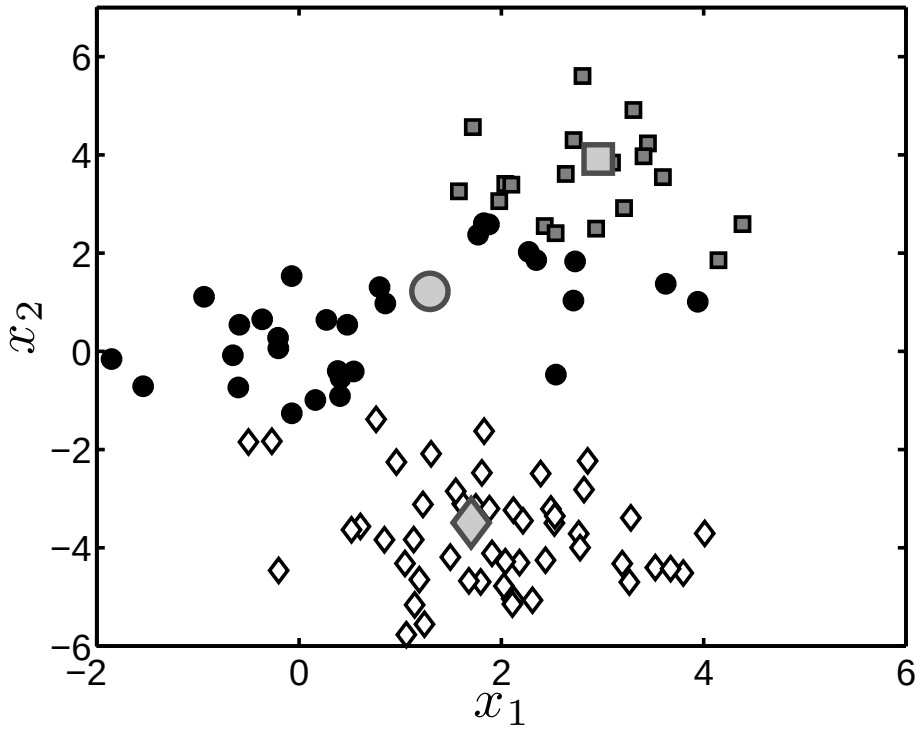
- Assign point to closest mean.

K-means – example



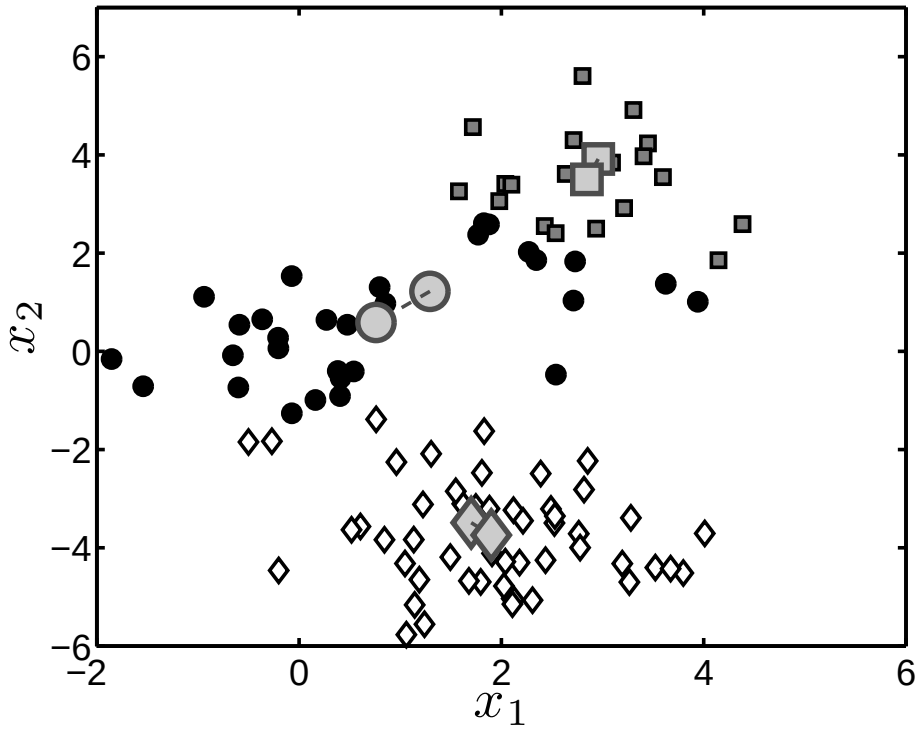
- Update mean.

K-means – example



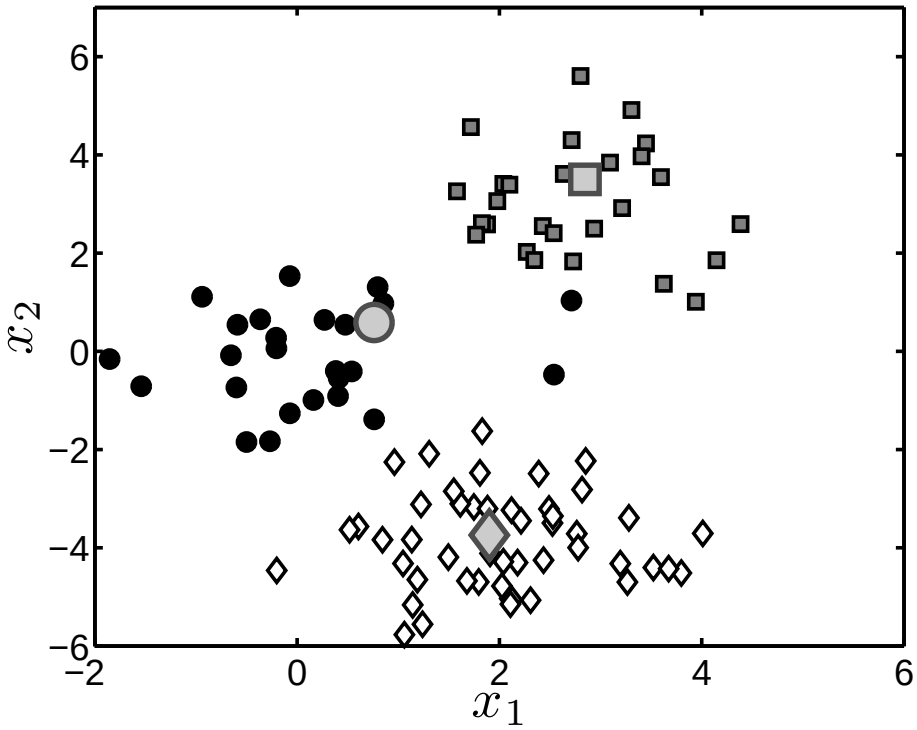
- Assign point to closest mean.

K-means – example



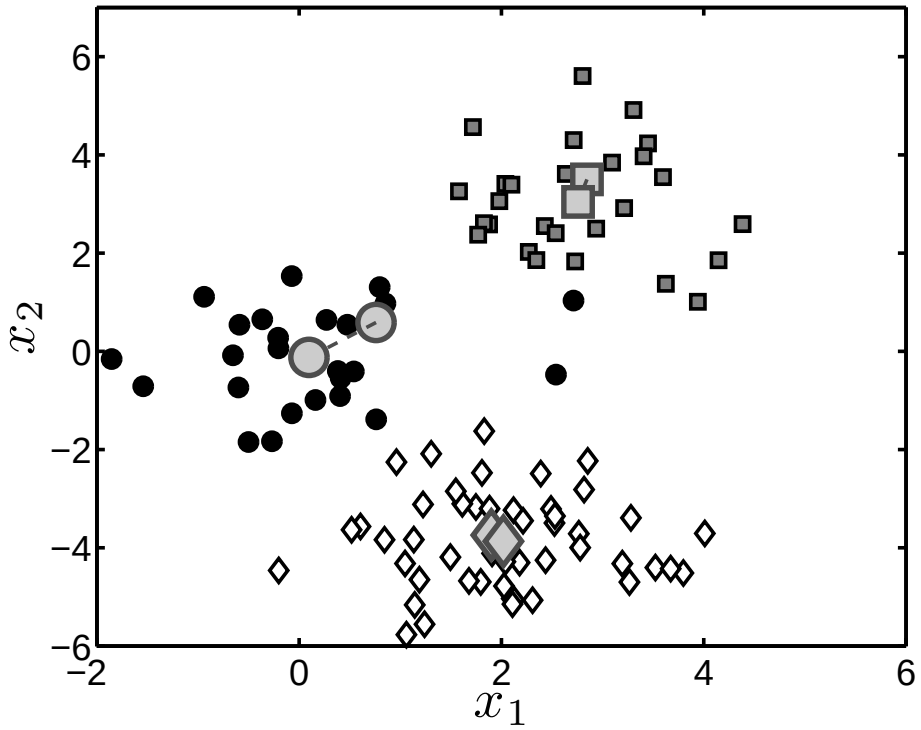
- Update mean.

K-means – example



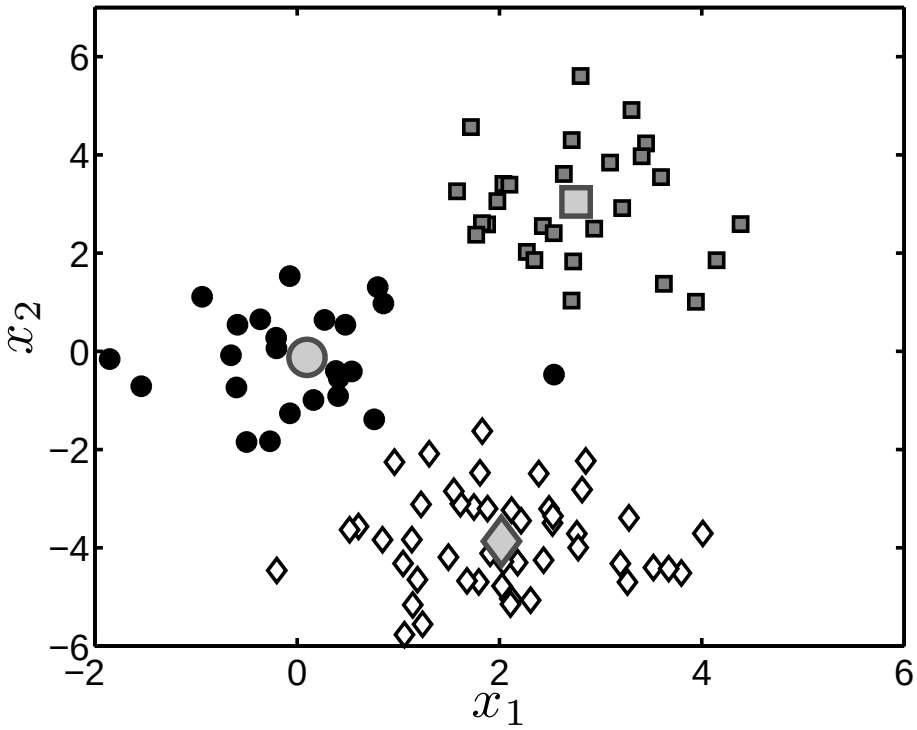
- Assign point to closest mean.

K-means – example



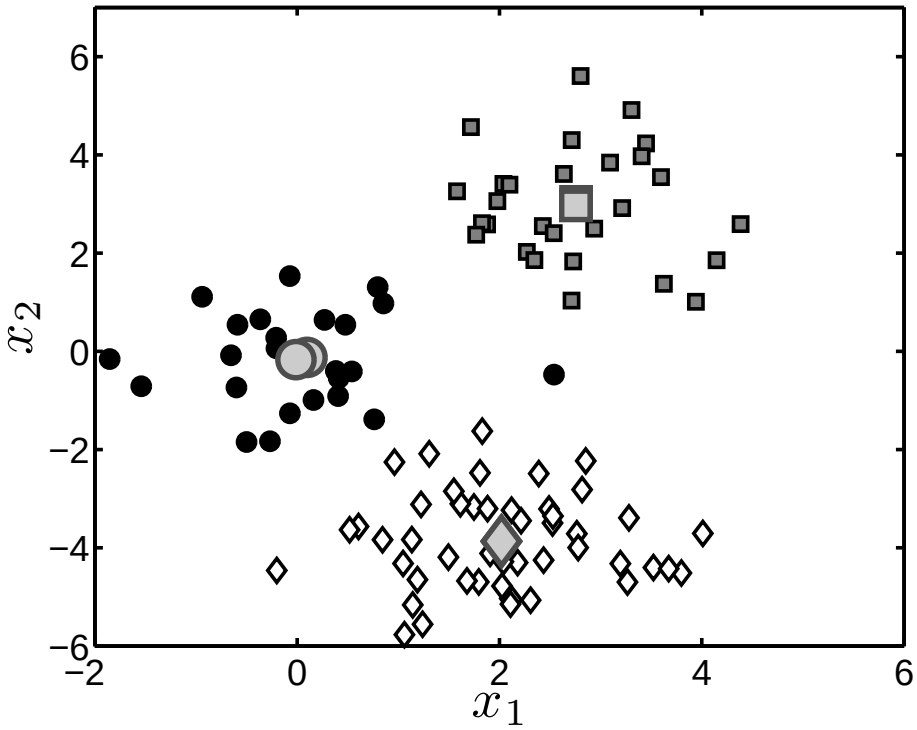
- Update mean.

K-means – example



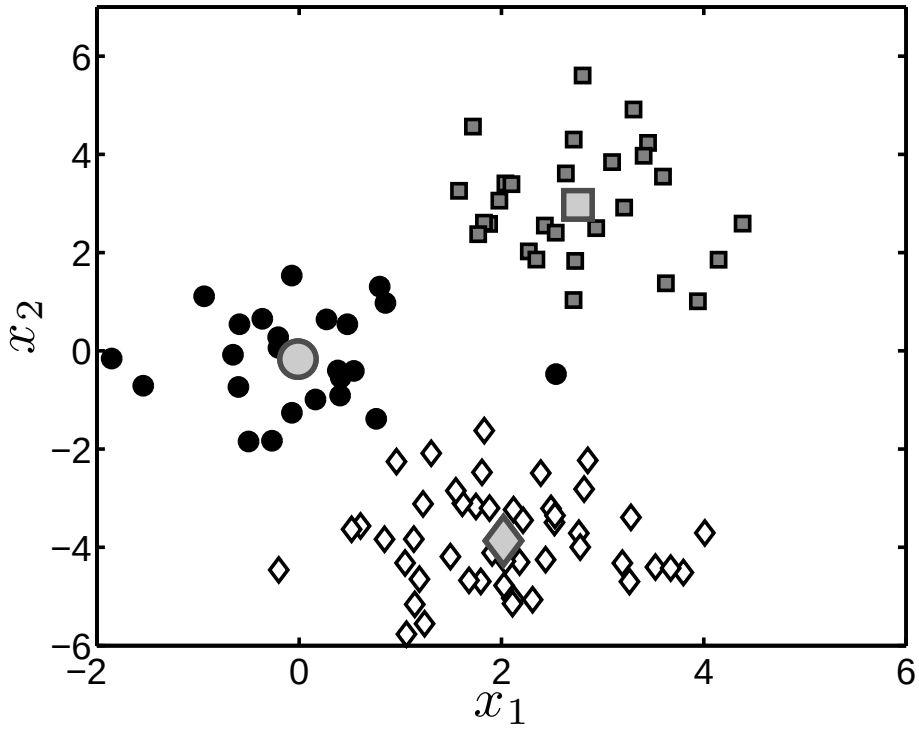
- Assign point to closest mean.

K-means – example



- Update mean.

K-means – example



- Solution at convergence.

Why K-means Converges

Menti-Quiz (Code: 7668-8743)

Explain very briefly why K-means converges

Why K-means Converges

Menti-Quiz (Code: 7668-8743)

Explain very briefly why K-means converges

- K-means minimizes the **intra-cluster distance** for each cluster.
- **Assignment step (optimize z_{nk}):**
- This choice minimizes squared distance for each point.
- So, this step **cannot increase** the objective.
- **Update step (optimize μ_k):**
- The optimal centroid is the mean of assigned points:
- Hence, also **cannot increase** the objective.

Variants of K-means

Menti-Quiz (Code: 7668-8743)

What is the best strategy to ensure faster convergence?

Variants of K-means

Menti-Quiz (Code: 7668-8743)

What is the best strategy to ensure faster convergence?

- Converges faster if centroids are away from each other.
- Indicates different regions of the data space and less number of reassignments during the update step.
- Algorithm called: **K-means++**.

Menti-Quiz (Code: 7668-8743)

Should we rethink the "update" step for sparse vectors?

Variants of K-means

Menti-Quiz (Code: 7668-8743)

What is the best strategy to ensure faster convergence?

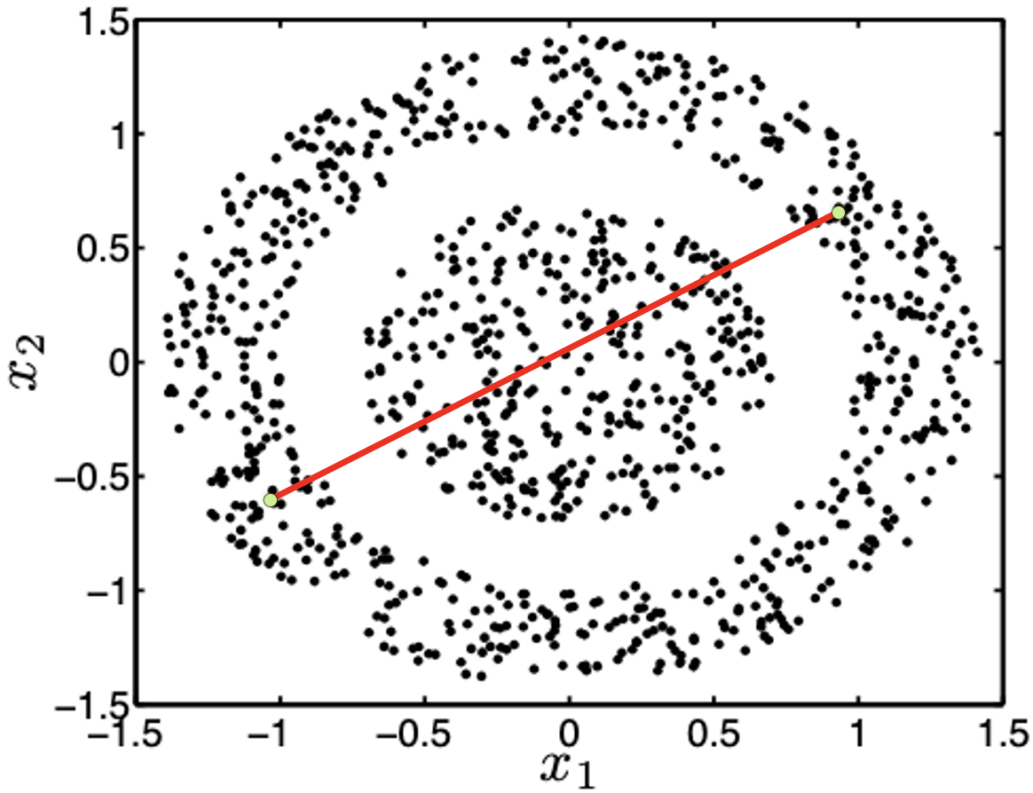
- Converges faster if centroids are away from each other.
- Indicates different regions of the data space and less number of reassignments during the update step.
- Algorithm called: **K-means++**.

Menti-Quiz (Code: 7668-8743)

Should we rethink the "update" step for sparse vectors?

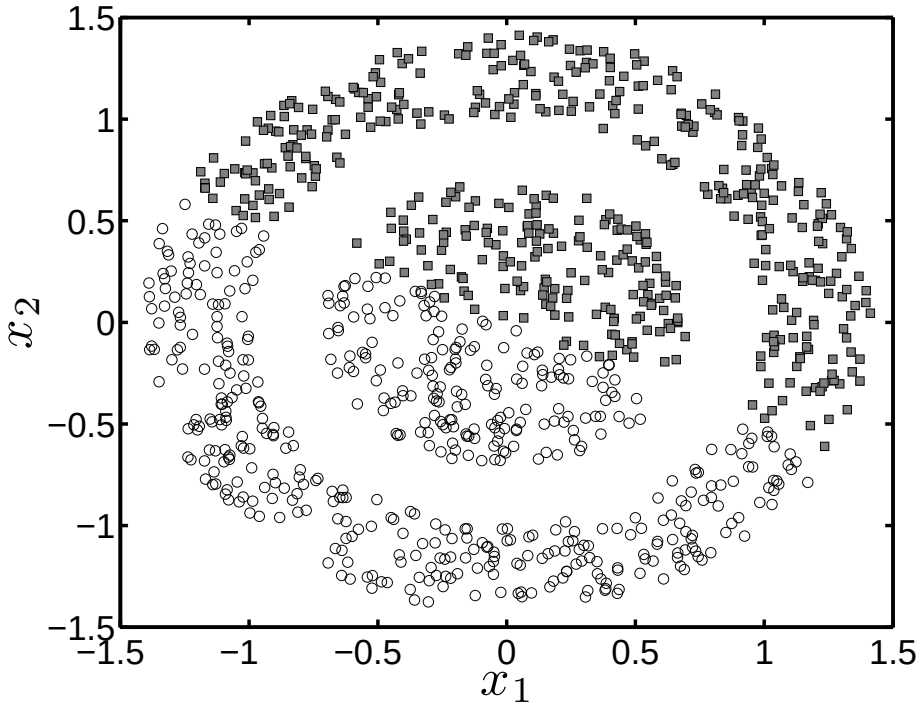
- Converges faster if centroids are restricted to be one of the data points.
- Adding two sparse vectors can **reduce sparsity**.
- Algorithm called: **K-medoids**.

When does K-means break?



- When the ideal clusters are non-convex in nature.
- What's a convex set?
  - $\forall \mathbf{x}, \mathbf{y} \in S, \lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in S, \forall \lambda \in [0,1]$.

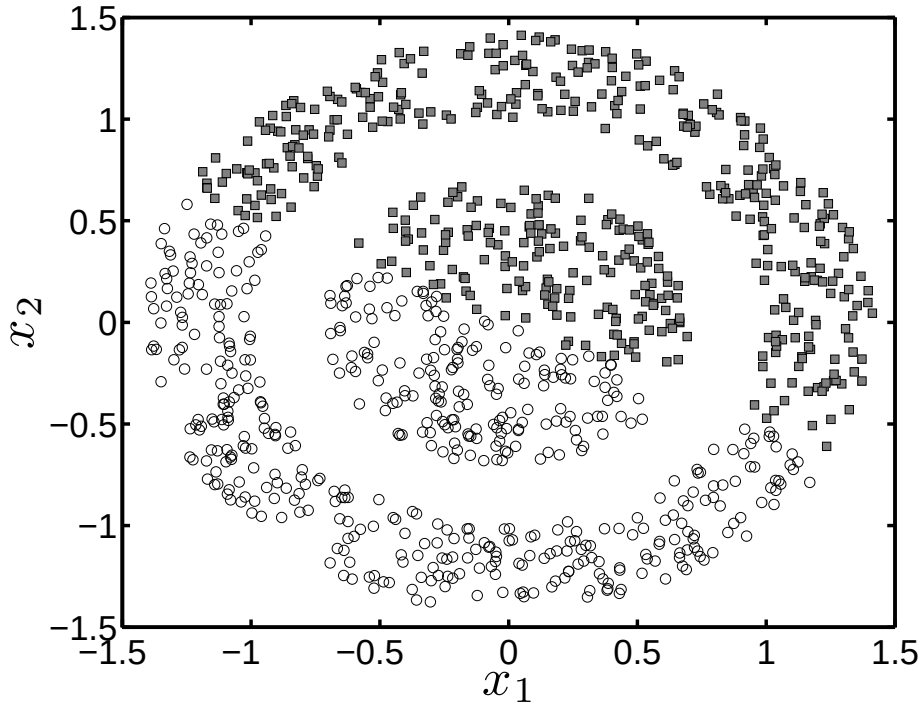
When does K-means break?



Menti-Quiz (Code: 7668-8743)

What's the problem with this type of data where the cluster pattern is obtained from a chaining effect?

When does K-means break?



Menti-Quiz (Code: 7668-8743)

What's the problem with this type of data where the cluster pattern is obtained from a chaining effect?

- Outer cluster can not be represented as a single point.

Kernelising K-means

- Maybe we can kernelise K-means?
- Distances:

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- Cluster means:

$$\boldsymbol{\mu}_k = \frac{\sum_{m=1}^{N} z_{mk}\mathbf{x}_m}{\sum_{m=1}^{N} z_{mk}} = \frac{\sum_{m=1}^{N} z_{mk}\mathbf{x}_m}{N_k}$$

- That's easy. Just substitute $\boldsymbol{\mu}_m = N_k^{-1} \sum_m z_{mk}\phi(\mathbf{x}_m)$

Menti-Quiz (Code: 7668-8743)

Unfortunately, this doesn't work. Why?

Kernelising K-means

- Maybe we can kernelise K-means?
- Distances:

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- Cluster means:

$$\boldsymbol{\mu}_k = \frac{\sum_{m=1}^{N} z_{mk}\mathbf{x}_m}{\sum_{m=1}^{N} z_{mk}} = \frac{\sum_{m=1}^{N} z_{mk}\mathbf{x}_m}{N_k}$$

- That's easy. Just substitute $\boldsymbol{\mu}_m = N_k^{-1} \sum_m z_{mk}\phi(\mathbf{x}_m)$

Menti-Quiz (Code: 7668-8743)

Unfortunately, this doesn't work. Why?

- We (often) don't know $\phi(\mathbf{x}_n)$!
- So we need to be a bit clever.

Kernelising K-means

- Rewrite distances as:

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_n - \boldsymbol{\mu}_k) = \left(\mathbf{x}_n - \frac{1}{N_k}\sum_{m=1}^{N} z_{mk}\mathbf{x}_m\right)^{\mathsf{T}} \left(\mathbf{x}_n - \frac{1}{N_k}\sum_{m=1}^{N} z_{mk}\mathbf{x}_m\right)$$

- $\mathbf{x}_n$: pivot point, $\mathbf{x}_m$s replace μ_k.
- **No pre-aggregation** as in K-means. Models the pivot's distance relative to every other point in that cluster.

Kernelising K-means

General distance formula

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_n - \boldsymbol{\mu}_k) = \left(\mathbf{x}_n - \frac{1}{N_k}\sum_{m=1}^{N} z_{mk}\mathbf{x}_m\right)^{\mathsf{T}} \left(\mathbf{x}_n - \frac{1}{N_k}\sum_{m=1}^{N} z_{mk}\mathbf{x}_m\right)$$

- Multiply:

$$\underbrace{\mathbf{x}_n^{\mathsf{T}}\mathbf{x}_n}_{\text{Quadratic term}} - \frac{2}{N_k}\sum_{m=1}^{N} \underbrace{z_{mk}\mathbf{x}_m^{\mathsf{T}}\mathbf{x}_n}_{\text{Pivot wrt others}} + \frac{1}{N_k^2}\sum_{m,l} \underbrace{z_{mk}z_{lk}\mathbf{x}_m^{\mathsf{T}}\mathbf{x}_l}_{\text{others wrt others}}$$

- Kernel substitution:

$$k(\mathbf{x}_n, \mathbf{x}_n) - 2N_k^{-1}\sum_{m=1}^{N} z_{mk}\, k(\mathbf{x}_n, \mathbf{x}_m) + N_k^{-2}\sum_{m,l=1}^{N} z_{mk}z_{lk}\, k(\mathbf{x}_m, \mathbf{x}_l)$$

Kernelized K-means Algorithm

- Choose a kernel and any necessary parameters.
- Start with random assignments z_{nk}.

Assigning to nearest centre

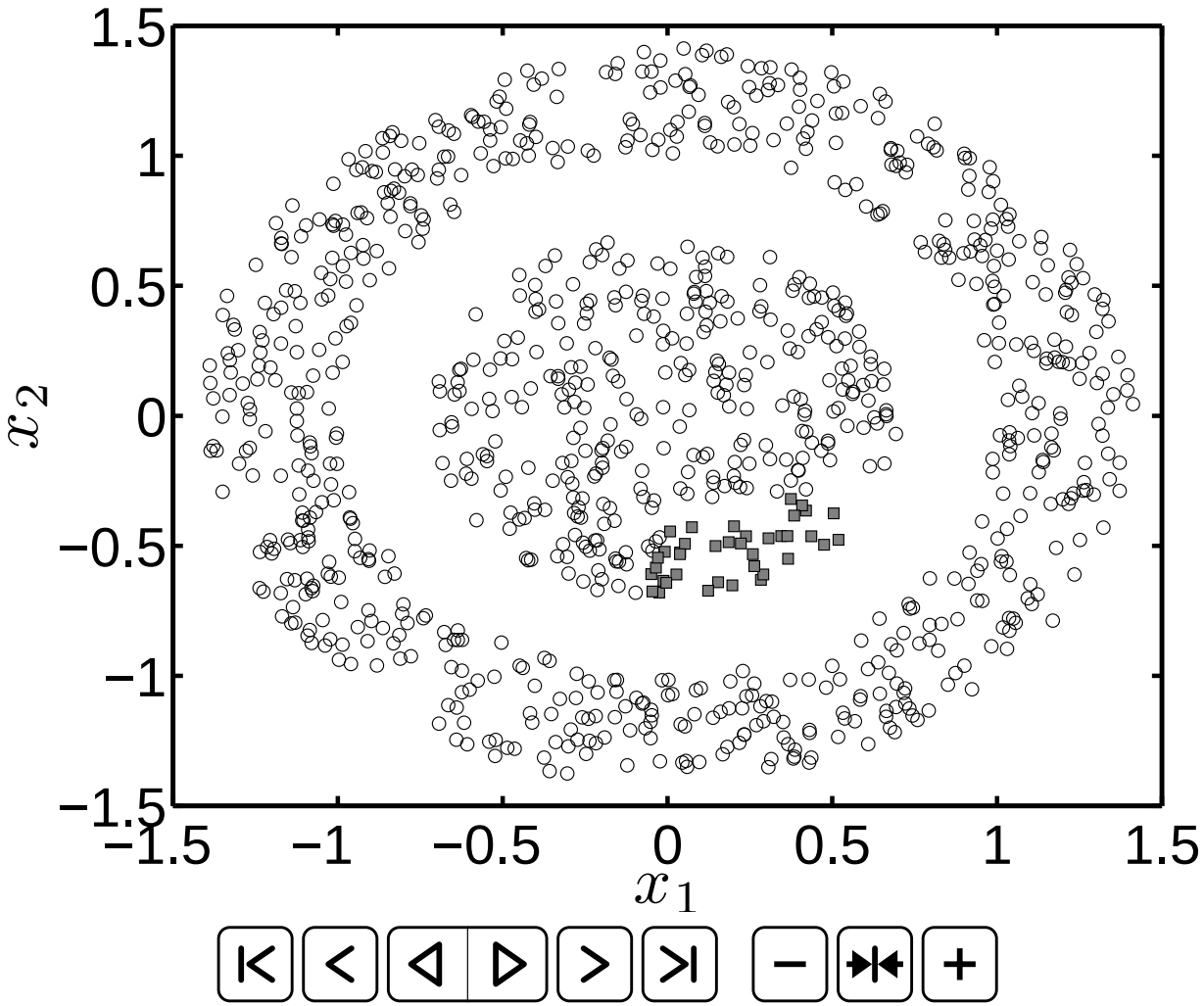
Find z_{nk} such that the following is minimised:

$$k(\mathbf{x}_n, \mathbf{x}_n) - 2N_k^{-1} \sum_{m=1}^{N} z_{mk}\, k(\mathbf{x}_n, \mathbf{x}_m) + N_k^{-2} \sum_{m,l=1}^{N} z_{mk} z_{lk}\, k(\mathbf{x}_m, \mathbf{x}_l)$$

- $k(\mathbf{x}_n, \mathbf{x}_n)$: Norm of pivot.
- $k(\mathbf{x}_n, \mathbf{x}_m)$: How close is pivot to other points in the cluster.
- $k(\mathbf{x}_m, \mathbf{x}_l)$: How compact is the cluster.

- If assignments have changed, repeat.

Kernel K-means – example



Kernel K-means

- Makes simple K-means algorithm more flexible.
- Very sensitive to initial conditions – lots of local optima.
- Computationally more expensive.

Menti-Quiz (Code: 7668-8743)

Think about another disadvantage of Kernelized K-means

Kernel K-means

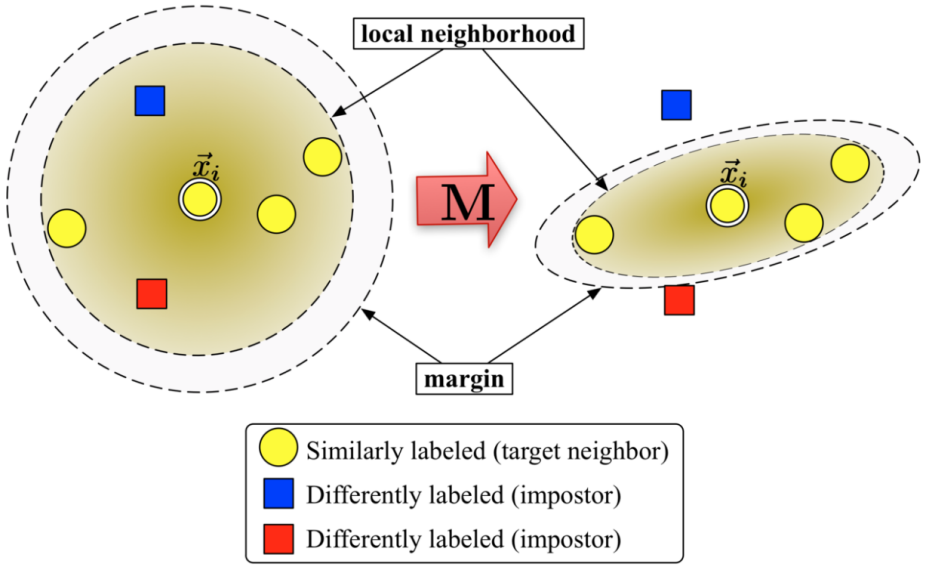
- Makes simple K-means algorithm more flexible.
- Very sensitive to initial conditions – lots of local optima.
- Computationally more expensive.

Menti-Quiz (Code: 7668-8743)

Think about another disadvantage of Kernelized K-means

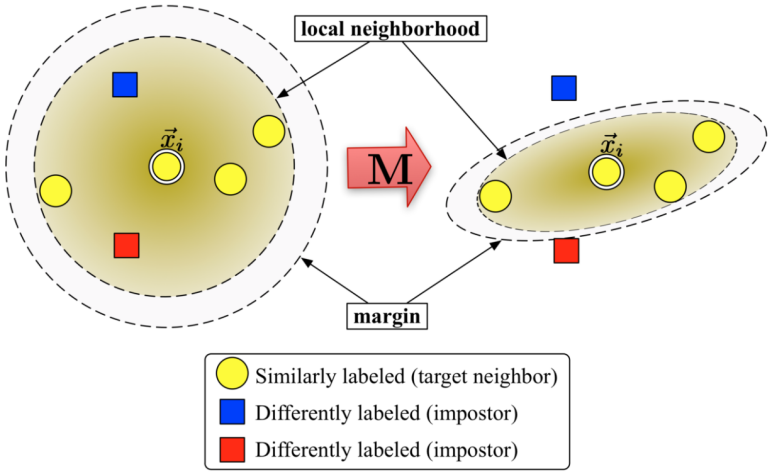
- Have to now set additional parameters for the kernel!
- Coefficient for Gaussian, or degree for polynomial kernel.

Parameterized similarity functions



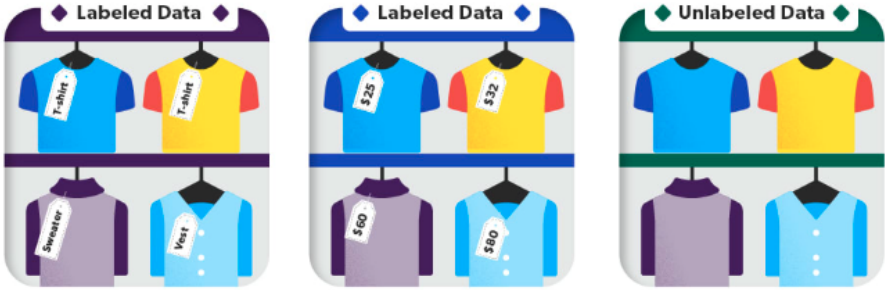
- Hand-engineered kernels $k(\mathbf{x}_i, \mathbf{x}_j)$ to parameterized similarity function $\theta(\mathbf{x}_i, \mathbf{x}_j)$.
- Need: labeled data in the form of **triples** $(\mathbf{x}, \mathbf{z}_p, \mathbf{z}_n)$.
  - $\mathbf{x}$ - a pivot point
  - $\mathbf{z}_p$ - a positive example (one that has the same class label as $\mathbf{x}$); note: we don't need to know the label itself.
  - $\mathbf{z}_n$ - a negative example (one that has a class label different from $\mathbf{x}$).

Noise Contrastive Estimation



- A simple approach is to apply **Logistic Regression** on **pairs** rather than on single instances.
- $\mathbf{x} \oplus \mathbf{z}_p$: Denotes concatenation of $\mathbf{x}$ and $\mathbf{z}_p$.
- Train parameters θ (binary cross-entropy) on two input pairs.
  - $\theta : \mathbf{x} \oplus \mathbf{z}_p \mapsto 1$ (pull the vector $\mathbf{z}_p$ closer to $\mathbf{x}$ in the parameterized space)
  - $\theta : \mathbf{x} \oplus \mathbf{z}_n \mapsto 0$ (push the vector $\mathbf{z}_n$ away from $\mathbf{x}$ in the parameterized space)

Parameterized Similarity in Kernelized K-means



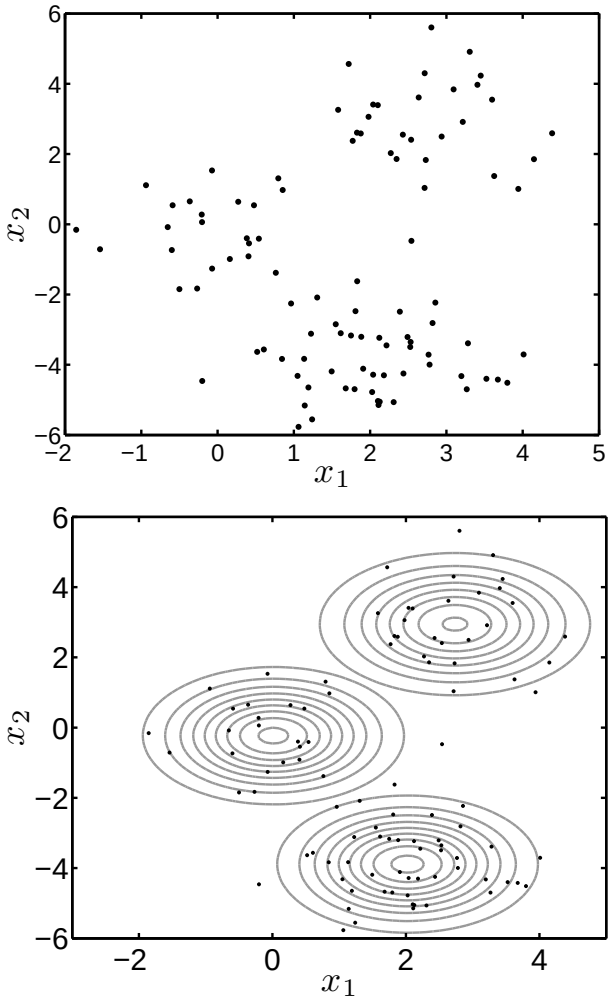
First learn the similarity space

- Use the domain-specific data to define which pairs are similar and which are not.
- Use this to train a logistic regression.
- $\theta(\mathbf{x}, \mathbf{z})$ now gives a sigmoid probability, which is a **parameterized similarity**.

Use as the kernel function in K-means

$$\theta(\mathbf{x}_n, \mathbf{x}_n) - 2N_k^{-1} \sum_{m=1}^{N} z_{mk}\, \theta(\mathbf{x}_n, \mathbf{x}_m) + N_k^{-2} \sum_{m,l=1}^{N} z_{mk} z_{lk}\, \theta(\mathbf{x}_m, \mathbf{x}_l)$$

Mixture models – thinking generatively



- Hypothesis: A model that could have created this data?
- Each $\mathbf{x}_n$ seems to have come from one of three (Gaussian) distributions?
- Once we estimate the Gaussians (easy to parameterize), we can get a clustering output.

Outline of a Generative Model for Clustering

- Δ: the parameters of all these distributions.
- Initially: Try some random configuration of Δ.
  - This is similar to choosing random cluster centroids in K-means!
- To generate $\mathbf{X}$:
- For each n:
  1. Pick one of the K components.
  2. Sample $\mathbf{x}_n$ from this distribution.
  3. For this point, we know its cluster label.
- Maximise the likelihood to estimate Δ given $\mathbf{X}$.
  - This is similar to updating the cluster centres in K-means!
  - Repeat

Underlying distribution - MoG

- Δ: for Gaussians is something that we have seen a lot of times before.
- $P(\mathbf{X}, \boldsymbol{\mu}, \Sigma) = \prod_{i=1}^{N} \sum_{k=1}^{K} \underbrace{P(z_n = k)}_{\text{Prior}} \underbrace{P(\mathbf{x}_n | \mu_k, \Sigma_k)}_{\text{Posterior}}$
- Parameters:
  - π_k (prior of component k)
  - μ_k, Σ_k: Centroid and spread of component k

Underlying distribution - MoG

- Guess initial values of π_k, μ_k, Σ_k.
- Repeat until convergence.

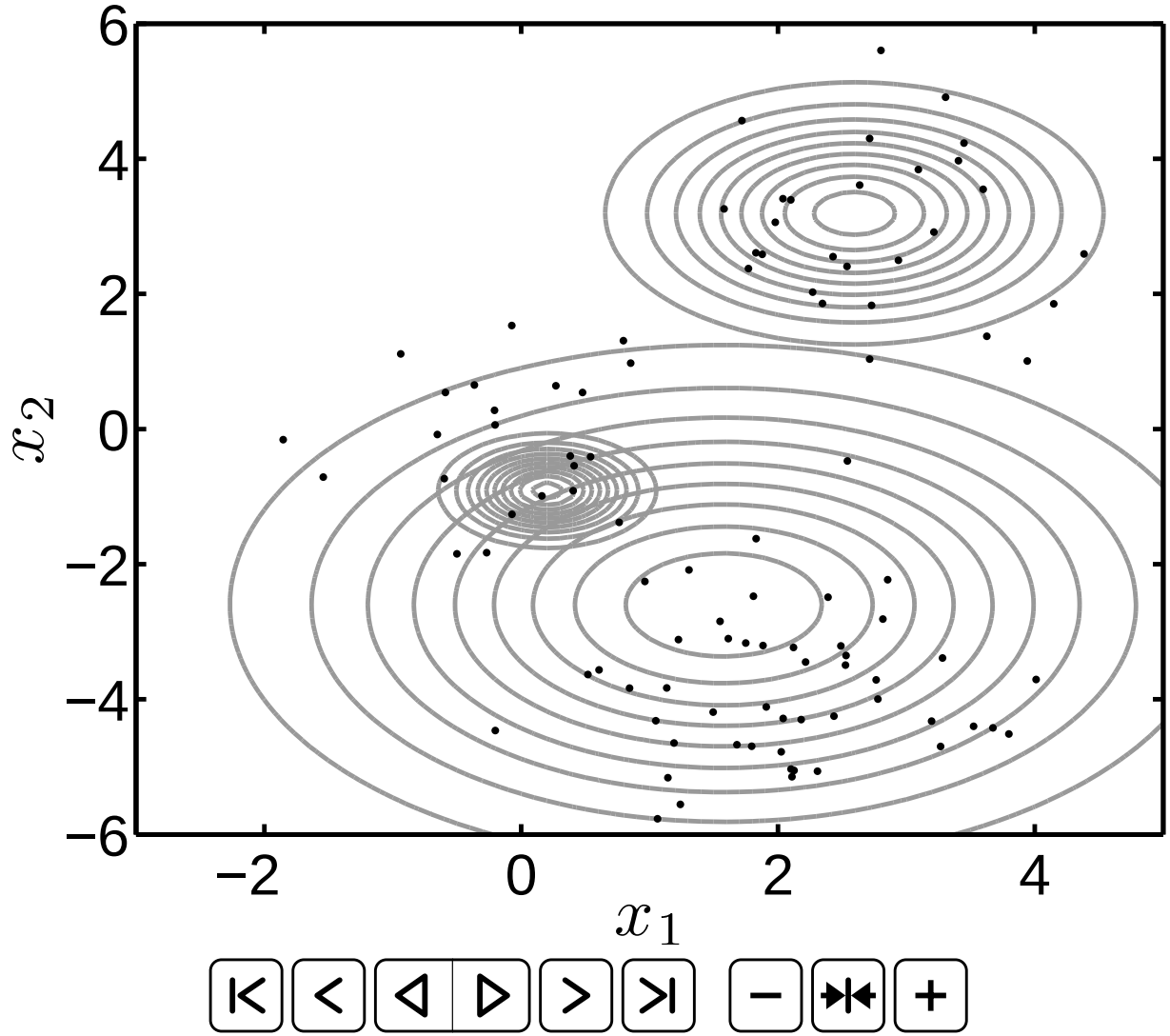
E-step ($\sim$ distance from k-th centroid in K-means)

$$P_{(i,k)} = P(z_i = k|\mathbf{x}_i) = \pi_k(\mathbf{x}_i - \mu_k)^T \Sigma_k (\mathbf{x}_i - \mu_k)$$

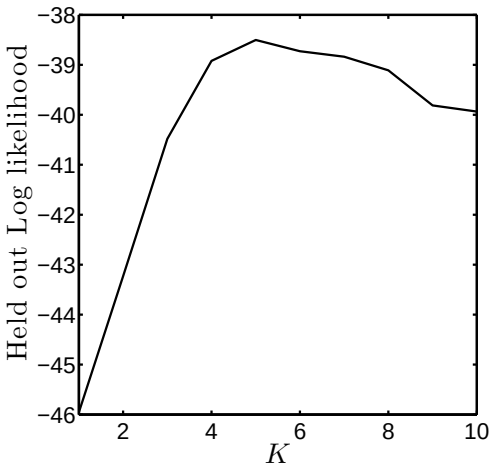
M-step ($\sim$ update centroids in K-means)

- $\mu_k \leftarrow \frac{\sum_{i=1}^{N} P_{(i,k)} \mathbf{x}_i}{\sum_{i=1}^{N} P_{(i,k)}}$ (Compute the expected value)
- $\mathbf{\Sigma}_k \leftarrow \frac{\sum_{i=1}^{N} P_{(i,k)} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^{N} P_{(i,k)}}$
- $\pi_k \leftarrow \frac{1}{N} P_{(i,k)}$

MoG Example



Problems with K-means (and similar) algorithms



- How do we know K?
  - #clusters in K-means.
  - #Gaussian components in EM algorithm.
- Optimize a criterion via cross-validation.
  - MoG: Maximize posterior likelihood for the observed data.
  - K-means: Minimize aggregate distance of each point from its cluster centre.
  - Minimize: $\frac{\text{Intra-cluster distance}}{\text{Inter-cluster distance}}$

Clustering Evaluation

Clustering Evaluation

Supervised model evaluation

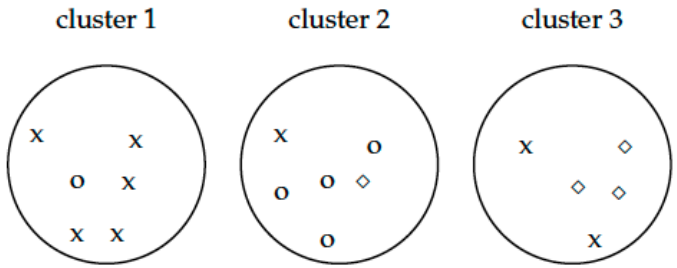
- Ground-truth labels for each instance.
- Compute how many times your predicted labels match the true labels - **Accuracy**.
- We also have looked at **class-wise precision** and **recall**.

Clustering Evaluation

- **Intrinsic**: Without ground-truth - Clustering Criterion.
- **Extrinsic**: With ground-truth (true class labels) - Purity, Rand-Index.

- Intrinsic Evaluation:
  - Ratio - not bounded in [0, 1].
  - No check against ground-truth.

Purity



Purity (Average Homogeneity of Clusters)

- $\Omega = \omega_1, \ldots, \omega_k$ - set of K clusters.
- $\mathcal{C} = C_1, \ldots, C_J$ - set of J class labels.

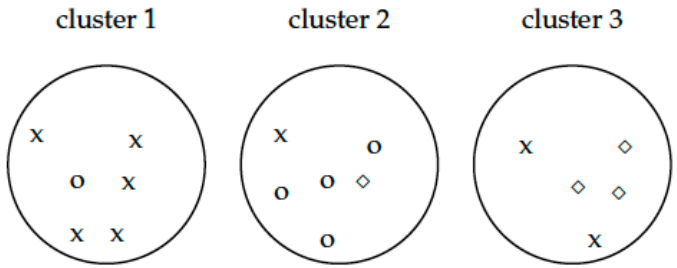
$\text{Purity}(\Omega, \mathcal{C}) = \frac{1}{N} \sum_{k=1}^{K} \max_{j=1}^{J} |\omega_k \cap C_j|$

- Mark each cluster by its **majority label**.
- Purity is $1/17 \times (5 + 4 + 3) = 0.71$.

Menti-Quiz (Code: 7668-8743)

What happens when K is too large?

Purity



Purity (Average Homogeneity of Clusters)

- $\Omega = \omega_1, \ldots, \omega_k$ - set of K clusters.
- $\mathcal{C} = C_1, \ldots, C_J$ - set of J class labels.

$\text{Purity}(\Omega, \mathcal{C}) = \frac{1}{N} \sum_{k=1}^{K} \max_{j=1}^{J} |\omega_k \cap C_j|$

- Mark each cluster by its **majority label**.
- Purity is $1/17 \times (5 + 4 + 3) = 0.71$.

Menti-Quiz (Code: 7668-8743)

What happens when K is too large?

Purity $\rightarrow 1$.

Rand Index (Pairwise accuracy)

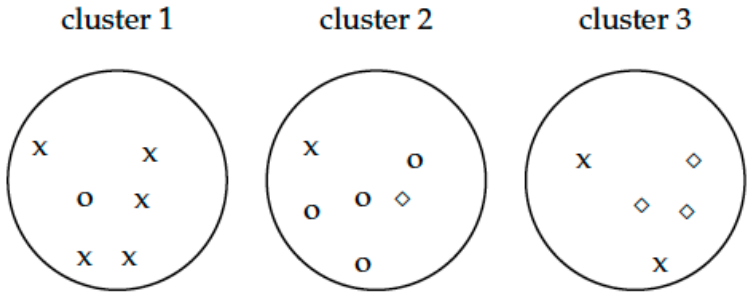
- Two partitions induced on the data:
  - Clustering (Predicted)
  - Class Labels (Ground-truth)
- **True positives**: **w** and **x** that have the same label (class) belong to the same cluster.
- **False positives**: **w** and **x** that have different labels are assigned to the same cluster.
- **False negatives**: **w** and **x** that have the same label are assigned to different clusters.
- **True negatives**: Two points with different class labels are assigned to different clusters.

Rand Index

$$RI = \frac{TP+TN}{N(N-1)/2}$$

Note: We can also define precision/recall/F-score this way.

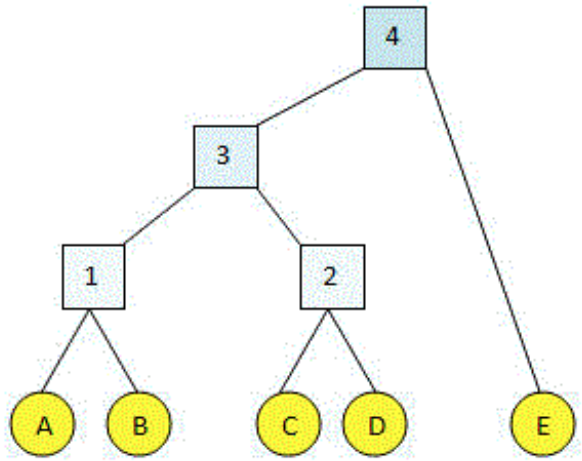
Rand Index

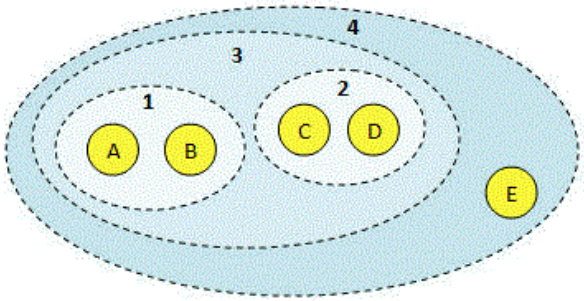


- Example computation:
  - $\text{TP}(X) = \text{TP}(X, C_1) + \text{TP}(X, C_2) + \text{TP}(X, C_3)$
  - $\text{TP}(X, C_1) = {}^5C_2$ (there are 5 Xs in cluster 1)
  - $\text{TP}(X, C_2) = 0$ (can't form a pair)
  - $\text{TP}(O, C_3) = 1$

Hierarchical Agglomerative Clustering

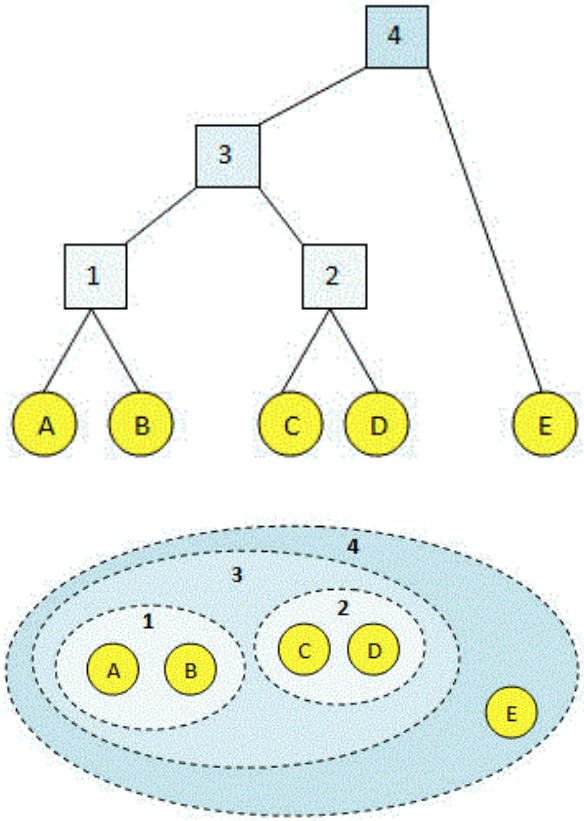
Hierarchical Agglomerative Clustering (HAC)





- K-means is a type of **flat clustering algorithm**.
  - What if we just executed K-means with $K = 5$?
  - If we want to see a different grouping with 10 clusters, we need to execute K-means again.
- HAC: Builds the necessary data structure to obtain a different grouping based on desired granularity.
- No need to execute the algorithm again.

Hierarchical Agglomerative Clustering



1. A bottom-up approach: starts by assuming that **each singleton point is a cluster of its own**.
2. Choose a pair of items X and Y that are **most similar** (**smallest distance**) from each other.
   - **Important**: X and Y are sets of points rather than single instances (obviously they also can be singleton sets).
3. Merge X with Y to form $Z = X \cup Y$.
4. Repeat the above.

How to compute distances between sets?

Different set similarity functions $\rightarrow$ different selection of sets to merge $\rightarrow$ different clustering output.

Menti-Quiz (Code: 7668-8743)

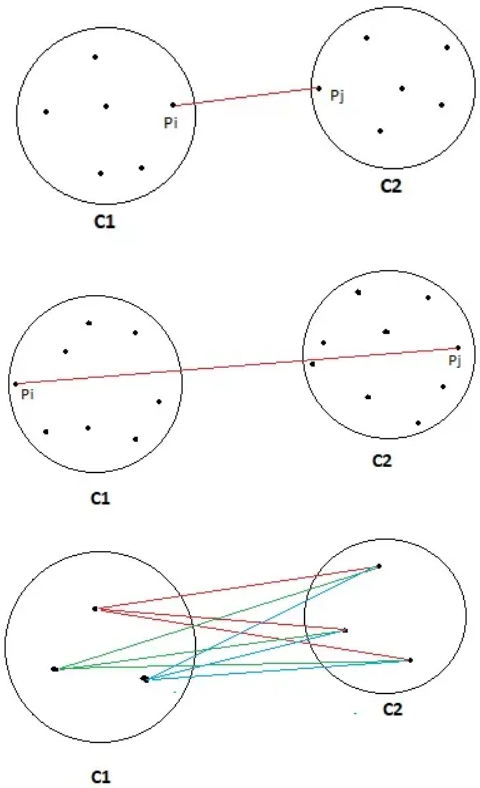
Write a short note on how to compute distances between two sets of points.

How to compute distances between sets?

Different set similarity functions $\rightarrow$ different selection of sets to merge $\rightarrow$ different clustering output.

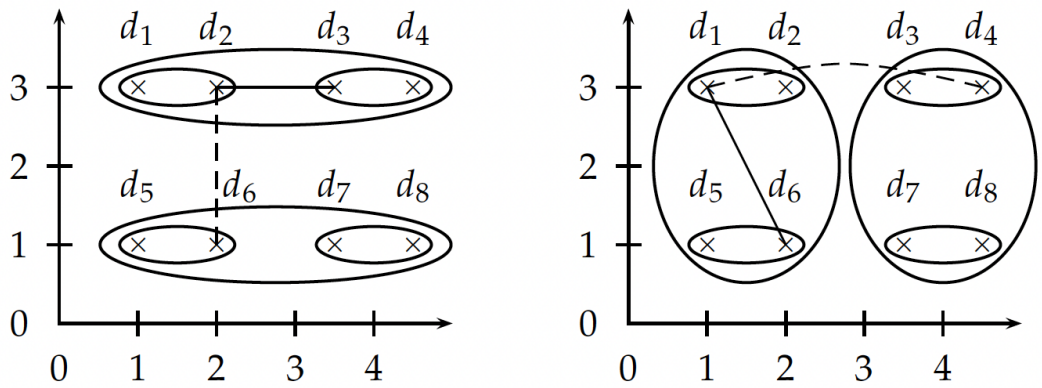
Menti-Quiz (Code: 7668-8743)

Write a short note on how to compute distances between two sets of points.



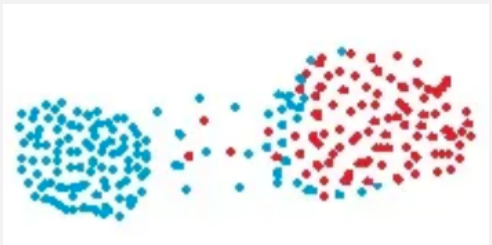
- **Single Link**:
  $d_{SL}(X, Y) = \min_{x \in X, y \in Y} d(x, y)$
- **Complete Link**:
  $d_{CL}(X, Y) = \max_{x \in X, y \in Y} d(x, y)$
- **Average Link**:
  $d_{AL}(X, Y) = \sum_{x \in X, y \in Y} d(x, y)$

Single-Link vs. Complete Link



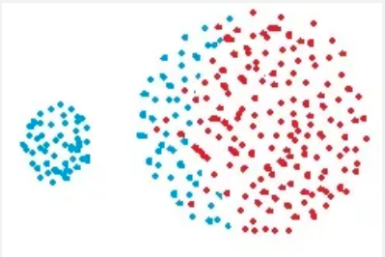
- Single Link: Produces a chaining effect.
- Complete Link: Leads to outliers.

More examples



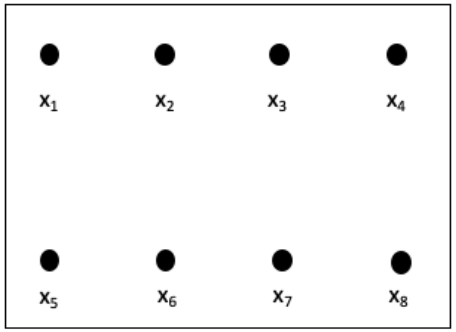
SL

CL



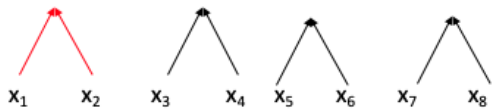
CL

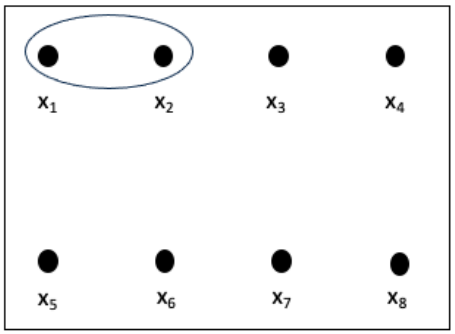
Example of HAC with Single Linkage



	x1	x2	x3	x4	x5	x6	x7	x8
x1		1	1	1	2	2.23	2.82	3.6
x2			1	2	2.23	2	2.23	2.82
x3								
x4								
x5								
x6								
x7								
x8								

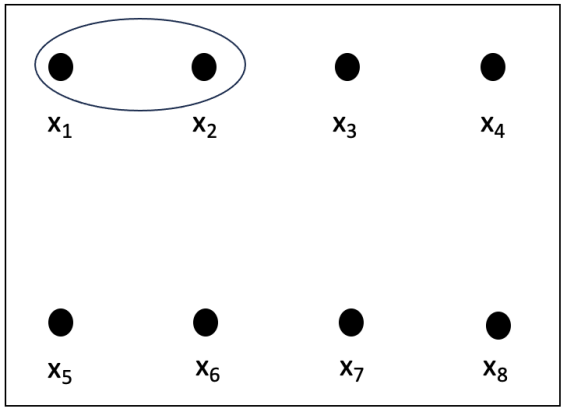
Candidate possibilities for merging two sets





- Select the two sets with minimum distance to merge.
- Initially:
  - Each set a singleton.
  - Many ties - resolve arbitrarily.

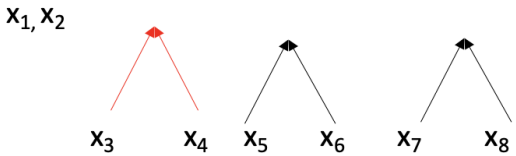
Example of HAC with Single Linkage

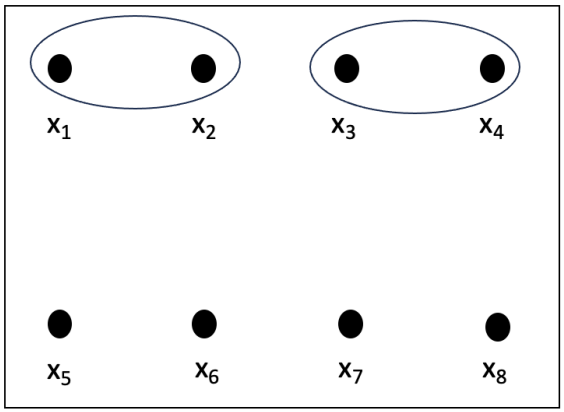


	x1, x2	x3	x4	x5	x6	x7	x8
x1, x2		1	?	?	?	?	?
x3							
x4							
x5							
x6							
x7							
x8							

Distance (X, Y) = minimum distance between all possible pairs of elements - one from X and the other from Y.

We could have merged x3 with {x1, x2}



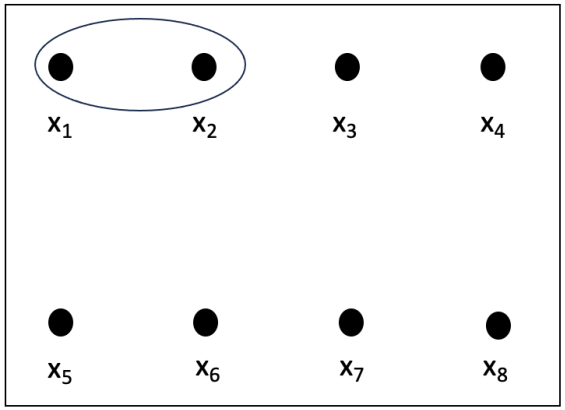


The distance matrix now changes because we have merged x_1 and x_2 in the previous step.

Menti-Quiz (Code: 7668-8743)

Compute distances of the new cluster from the remaining points.

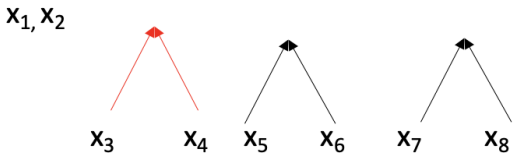
Example of HAC with Single Linkage

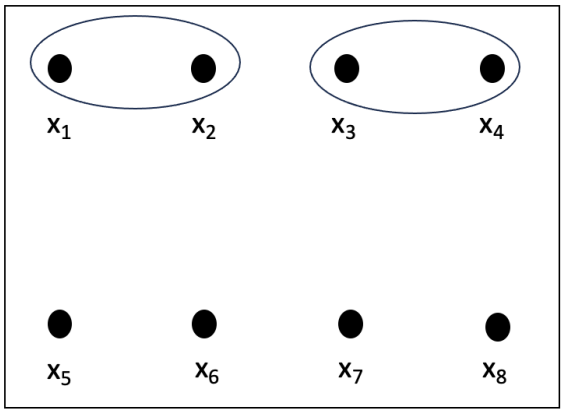


	x1, x2	x3	x4	x5	x6	x7	x8
x1, x2		1	?	?	?	?	?
x3							
x4							
x5							
x6							
x7							
x8							

Distance (X, Y) = minimum distance between all possible pairs of elements - one from X and the other from Y.

We could have merged x3 with {x1, x2}



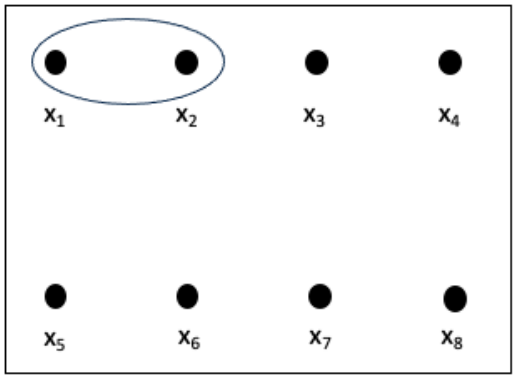


The distance matrix now changes because we have merged x_1 and x_2 in the previous step.

Menti-Quiz (Code: 7668-8743)

Compute distances of the new cluster from the remaining points.

Example of HAC with SL

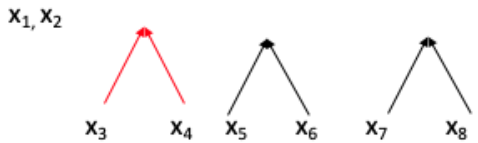


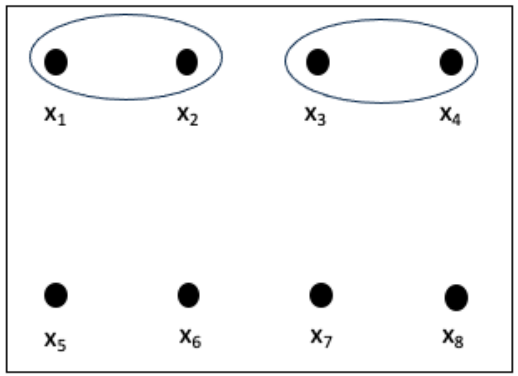
	x1, x2	x3	x4	x5	x6	x7	x8
x1, x2		1	2	2	2	2.23	2.82
x3							
x4							
x5							
x6							
x7							
x8							

Why?

Distance (X, Y) = minimum distance between all possible pairs of elements - one from X and the other from Y.

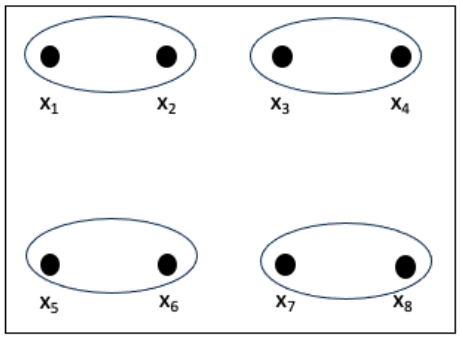
We could have merged x3 with {x1, x2}





- $d(\{x_1, x_2\}, x_3) = \min(d(x_1, x_3), d(x_2, x_3)) = \min(2, 1) = 1.$
- Continue like this till we have 4 clusters - $\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}$.

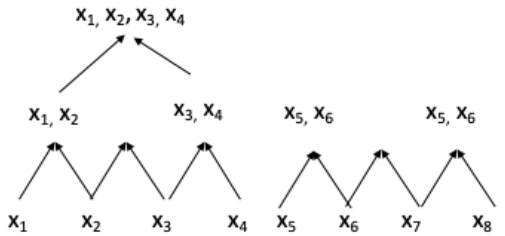
Example of HAC with SL

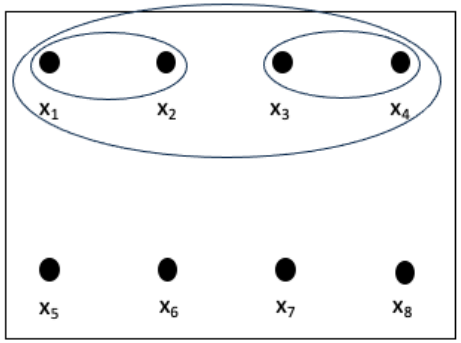


	x1, x2	x3, x4	x5, x6	x7, x8
x1, x2		1	2	2.23
x3, x4				
x5, x6				
x7, x8				

Now we need to compute distances between sets

After 4 merges





- Now no set is a singleton.
- See how we combine the sets $\{x_1, x_2\}$ with $\{x_3, x_4\}$ **and not with** $\{x_5, x_6\}$.
- **Homework**: Compute the SL distance between the two sets. What would be the CL distance?

Stopping Criterion for HAC

- If a desired number of clusters is obtained.
- From the tree constructed (merge history) take the partition which minimizes the clustering criteria (Inter-cluster distance divided by the Intra-cluster distance).

Clustering criterion

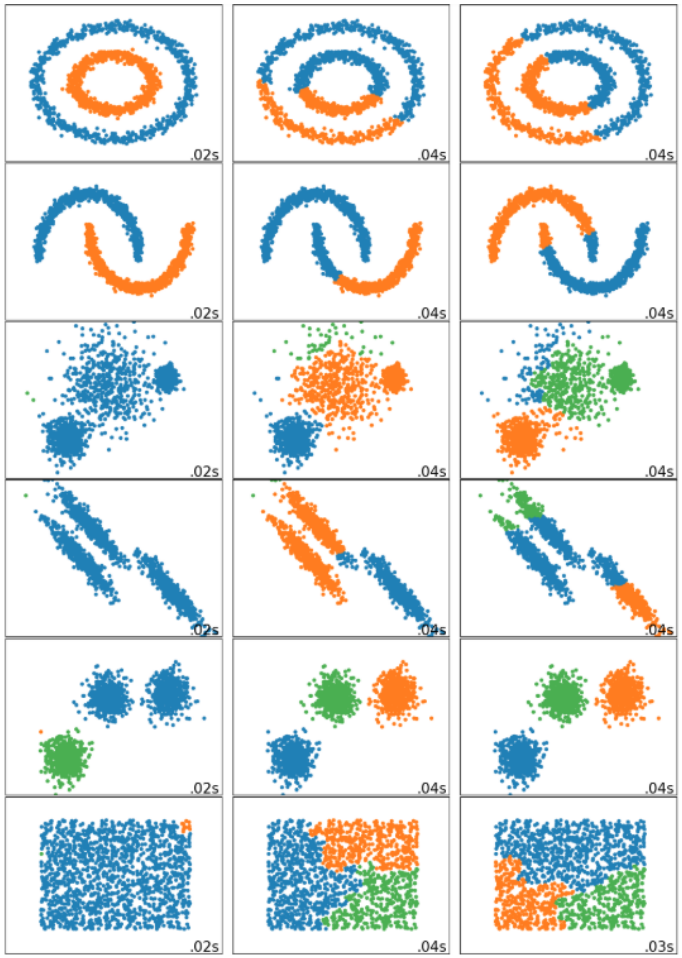
$$J(X,\Omega) = \frac{\sum_{\mathbf{x}\in X,\mathbf{y}\in X:\omega(x)\neq\omega(y)} d(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{x}\in X,\mathbf{y}\in X:\omega(x)=\omega(y)} d(\mathbf{x},\mathbf{y})}$$

- The gains in the clustering criterion falls below a threshold (a pre-set parameter).

No significant change in the partitions

$$\frac{J(X,\Omega_{i+1})}{J(X,\Omega_i)} < \tau$$

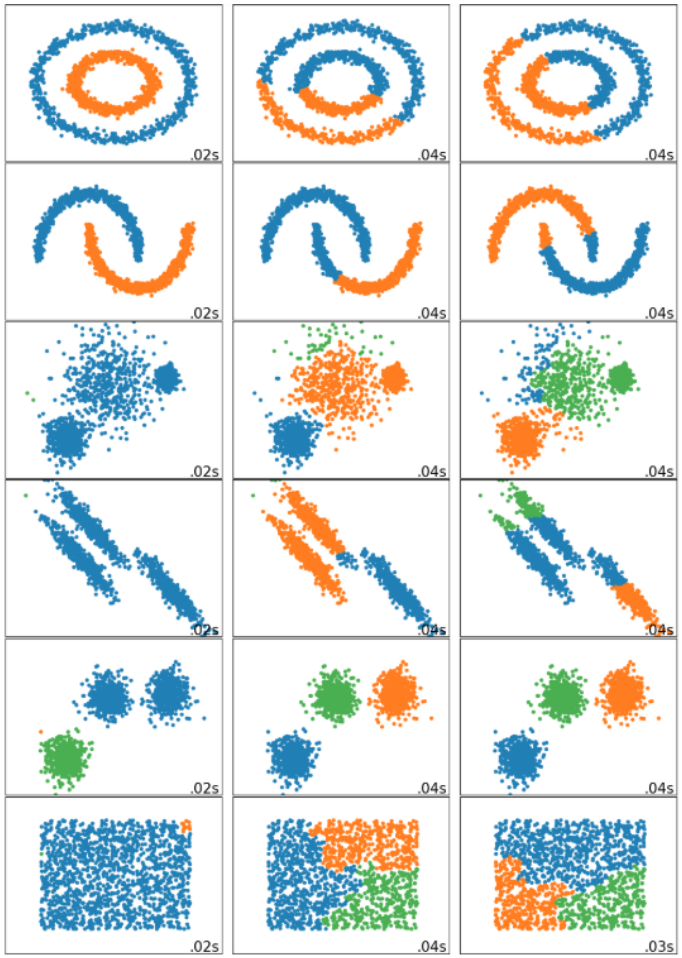
More Examples



Menti-Quiz (Code: 7668-8743)

Which linkage used?

More Examples



Menti-Quiz (Code: 7668-8743)

Which linkage used?

1. SL works well for the first two datasets (similar to Kernelized K-means with Gaussian kernel).
2. AL works similar to K-means (Homework: Why?)
3. See how CL yields **outlier clusters** (Homework: Why?).

Concluding words

Summary

- We covered **Clustering** – an unsupervised approach; quite different from what we had studied for the past 4 weeks.
- **K-means** - the simplest possible clustering approach.
- We now know how to **evaluate clustering**.
- **Kernelization** – work with generalised notions of similarities with kernel functions.
- Generalization of K-means via **MoG**.
- Hierarchical clustering — pros and cons

Final words!

That's the ML course!

Thank you all! All the best for your exams!

PhD Advertisement

- Drop me a message on Teams if you want to do a PhD with me on **Agentic Multi-modal Models**!
- You need to submit a **research statement** on a **novel and interesting problem**.
- **Scholarship opportunities**:
  - School scholarship — easier for home students, very tough for international students. **Deadline** is **31st Jan'26**.
  - Apply to a CDT: 🔗 check the deadline on page.