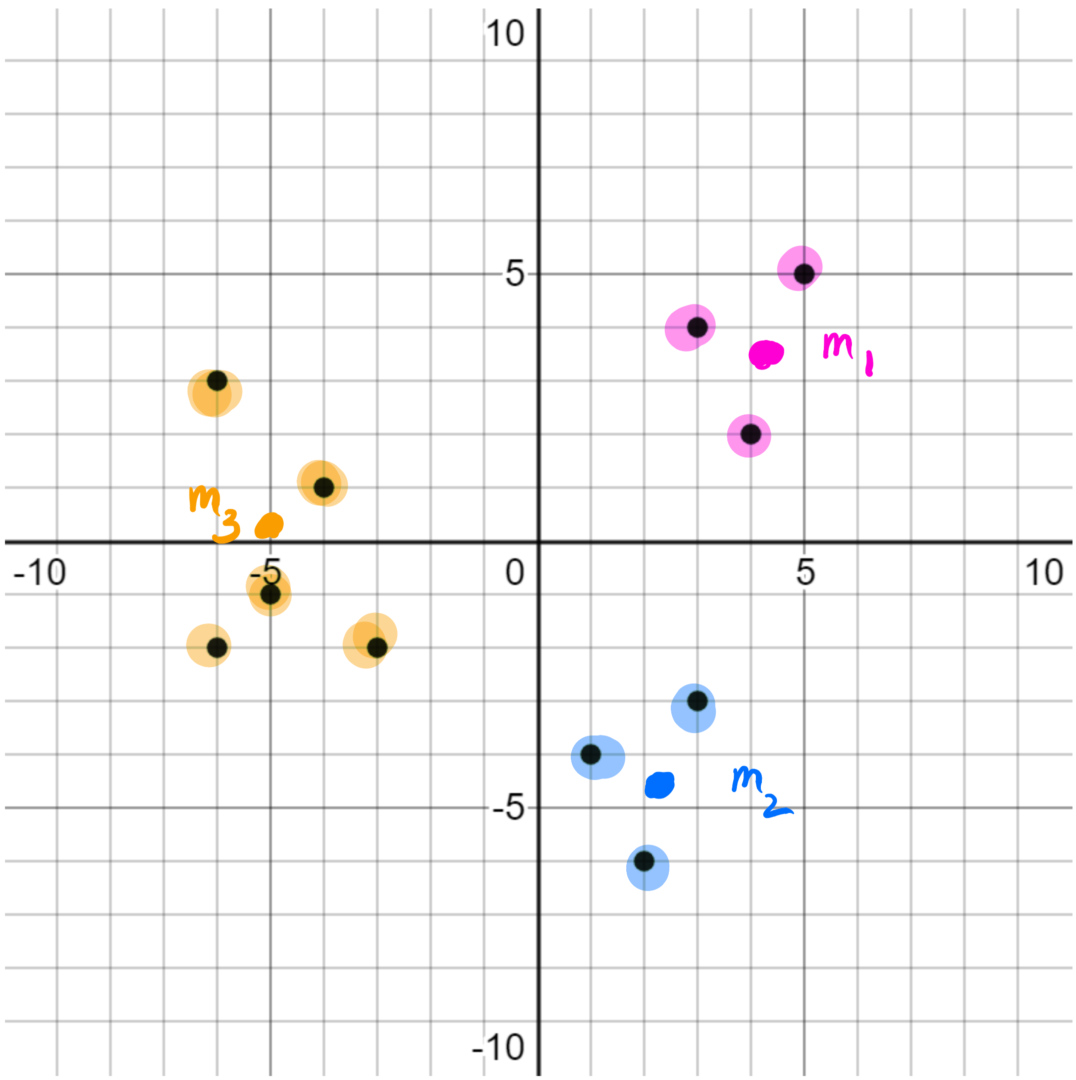
Lecture 17: k-means

- clustering algorithm
- unsupervised learning
- input data: d-dimensional vectors, with numerical entries
- algorithm clusters data into k similar clusters.
- clusters are identified by their centroids (means)



$k = 3$

k-means Algorithm:

Let $\mathcal{D} = \{\mathbf{x}_1 \dots, \mathbf{x}_N\}$ be the data set, with d-dimensional input.
Fix k = the number of clusters, with $k \leq N$.

1. Start with a set of k-means in d-dimensional space

$$\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_k.$$

2. **Assign** each point to the mean to which it is closest: for $1 \leq j \leq k$

clusters: $S_j = \{\mathbf{x} \in \mathcal{D} : ||\mathbf{x} - \mathbf{m}_j|| \leq ||\mathbf{x} - \mathbf{m}_i|| \text{ for all } 1 \leq i \leq k\}$.

3. If there are changes in clustering assignments, **update** the means:

$$\mathbf{m}_j = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} \mathbf{x},$$

for $1 \leq j \leq k$, then go to step 2.

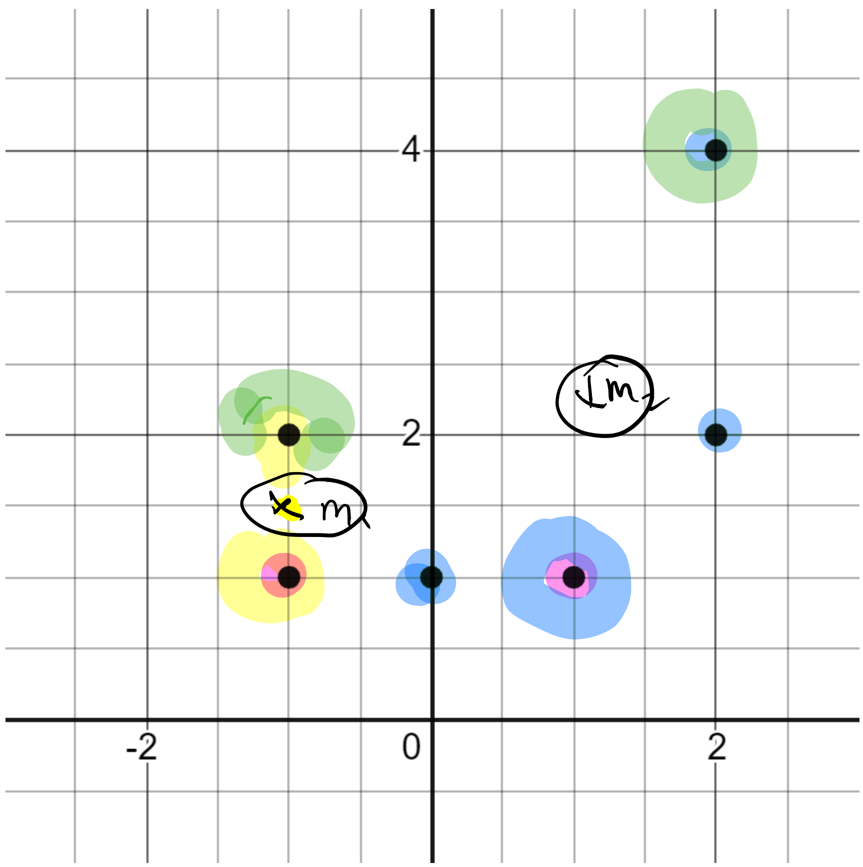
4. Stop if there are NO changes in clustering assignments.

5. Output the k-means $\mathbf{m}_j$ and the corresponding clusters S_j.

How to initialize:

- Forgy initialization: pick k values at random from $\mathcal{D}$ for the initial $\mathbf{m}_j$ $(1 \leq j \leq k)$
- Random partition: cluster $\mathcal{D}$ into k sets, and compute the means as the initial $\mathbf{m}_j$ $(1 \leq j \leq k)$
- Maximin: choose the first centroid at random. For $j > 1$, pick the j-th centroid by choosing the point from the data set for which the minimum distance to previously picked centroids is largest. This way, the centroids are far from each other.

$k=2$



Choosing k:

maximize

- Most of the time, k is forced from the context.
- If one can choose the best k, one can consider the function

$\boxed{\frac{f'(k) - f'(k+1)}{1}}$

$$f(k) = \sum_{i=1}^{N} \|\mathbf{x}_i - m(\mathbf{x}_i)\|^2,$$

$f(N) = 0$

$f(1)$ is max

where $m(\mathbf{x}_i)$ stands for the centroid of the data point $\mathbf{x}_i$. Think of it as a measure of "error" in clustering.

bend

We choose the k where $f(k)$ "bends", that is, the k that makes the largest impact: it is large enough to capture difference in the clusters, yet it is not too large to overfit. ⇒ largest change in slope!

Remarks:

1. There is no training phase for this algorithm, so k-means is an unsupervised learning algorithm.
2. This algorithm may not lead to an *optimal* clustering.
3. Starting with different initial means may lead to different clusters.
4. Note that $f(k) = 0$ for $k = N$, so that would lead to no error in clustering, but it overfits, does no clustering at all.
5. To visualize the clustering algorithm, try the following websites:

`http://stanford.edu/class/ee103/visualizations/kmeans`

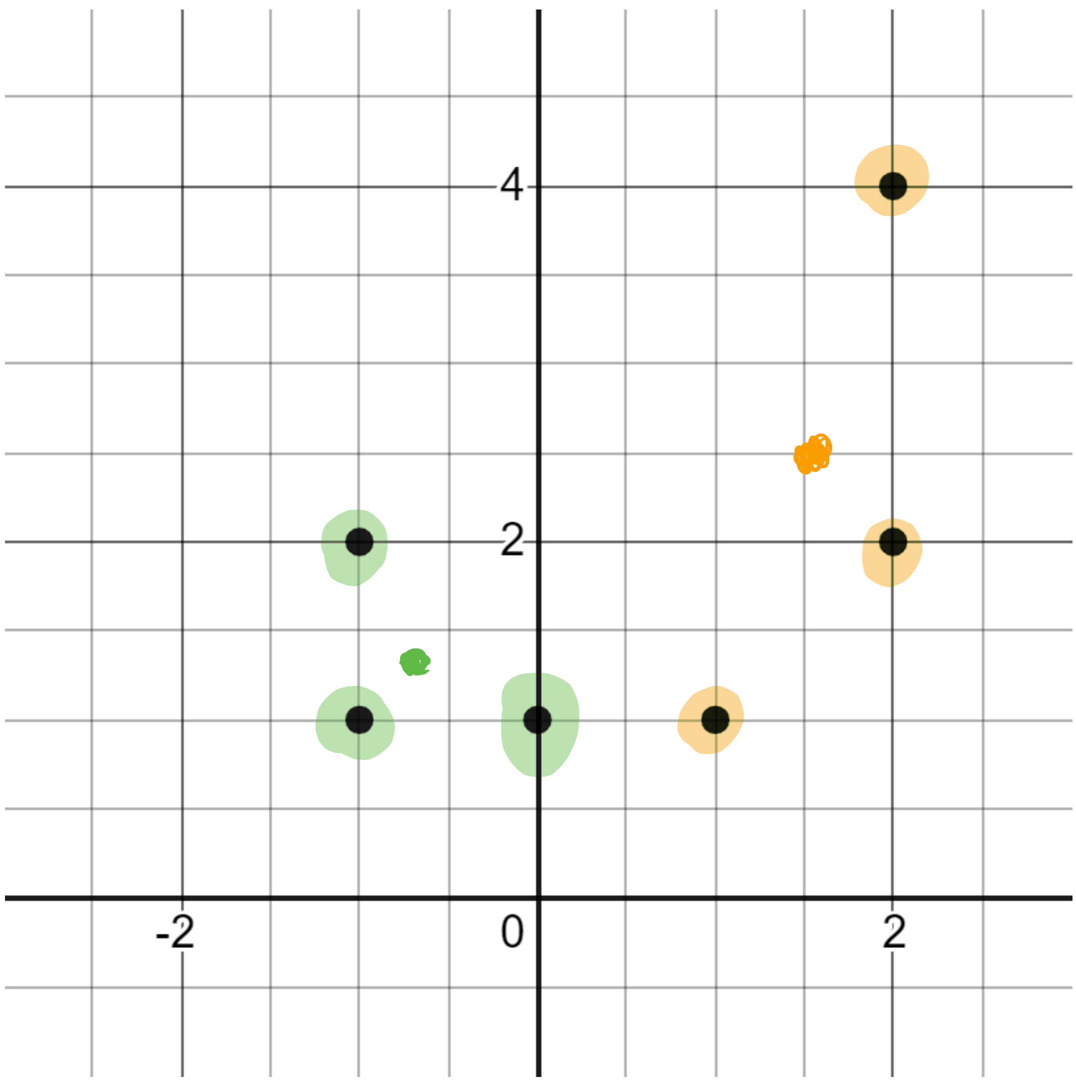
`https://www.naftaliharris.com/blog/visualizing-k-means-clustering`

Example

Let us consider the data set

$$\mathcal{D} = \left\{ \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \end{bmatrix} \right\}.$$

We will look for 2 clusters, by running through the 2-means algorithm.



- Let $\overset{S_1}{\mathbf{m}_1} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, $\overset{S_2}{\mathbf{m}_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- Compute the distance to the means and assign to the two clusters:

data point	dist^2 to $\mathbf{m}_1$	dist^2 to $\mathbf{m}_2$	cluster
$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$	0	$4+0$	S_1
$\begin{bmatrix} -1 \\ 2 \end{bmatrix}$	$0+1$	$4+1$	S_1
$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$1+0$	$1+0$	S_2 ← pick at random
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$4+0$	0	S_2
$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	$9+1$	$1+1$	S_2
$\begin{bmatrix} 2 \\ 4 \end{bmatrix}$	$9+9$	$1+9$	S_2

- Recompute the means:

$$\mathbf{m}_1 = \frac{1}{2}\left(\begin{bmatrix}-1\\1\end{bmatrix} + \begin{bmatrix}-1\\2\end{bmatrix}\right) = \begin{bmatrix}-1\\3/2\end{bmatrix}$$

$$\mathbf{m}_2 = \frac{1}{4}\left(\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}2\\2\end{bmatrix} + \begin{bmatrix}2\\4\end{bmatrix}\right) = \begin{bmatrix}5/4\\2\end{bmatrix}$$

- Compute the distance to the means and assign to the two clusters:

data point	dist^2 to $\mathbf{m}_1$	dist^2 to $\mathbf{m}_2$	cluster
$\begin{bmatrix}-1\\1\end{bmatrix}$	$0+\frac{1}{4}$	$\frac{81}{16}+1$	S_1
$\begin{bmatrix}-1\\2\end{bmatrix}$	$0+\frac{1}{4}$	$\frac{81}{16}+0$	S_1
moved $\longrightarrow$ $\begin{bmatrix}0\\1\end{bmatrix}$	$1+\frac{1}{4}$	$\frac{25}{16}+1$	S_1
$\begin{bmatrix}1\\1\end{bmatrix}$	$4+\frac{1}{4}$	$\frac{1}{16}+1$	S_2
$\begin{bmatrix}2\\2\end{bmatrix}$	$9+\frac{1}{4}$	$\frac{9}{16}+0$	S_2
$\begin{bmatrix}2\\4\end{bmatrix}$	$9+\frac{25}{4}$	$\frac{9}{16}+4$	S_2

- Recompute the means:

$$\mathbf{m}_1 = \frac{1}{3}\left(\begin{bmatrix}-1\\1\end{bmatrix}+\begin{bmatrix}-1\\2\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}-2/3\\4/3\end{bmatrix}$$

$$\mathbf{m}_2 = \frac{1}{3}\left(\begin{bmatrix}1\\1\end{bmatrix}+\begin{bmatrix}2\\2\end{bmatrix}+\begin{bmatrix}2\\4\end{bmatrix}\right) = \begin{bmatrix}5/3\\7/3\end{bmatrix}$$

- Compute the distance to the means and assign to the two clusters:

data point	dist^2 to $\mathbf{m}_1$	dist^2 to $\mathbf{m}_2$	cluster
$\begin{bmatrix}-1\\1\end{bmatrix}$	$\frac{1}{9}+\frac{1}{9}$	$\frac{64}{9}+\frac{16}{9}$	S_1
$\begin{bmatrix}-1\\2\end{bmatrix}$	$\frac{1}{9}+\frac{4}{9}$	$\frac{64}{9}+\frac{1}{9}$	S_1
$\begin{bmatrix}0\\1\end{bmatrix}$	$\frac{4}{9}+\frac{1}{9}$	$\frac{25}{9}+\frac{16}{9}$	S_1
$\begin{bmatrix}1\\1\end{bmatrix}$	$\frac{25}{9}+\frac{1}{9}$	$\frac{4}{9}+\frac{16}{9}$	S_2
$\begin{bmatrix}2\\2\end{bmatrix}$	$\frac{64}{9}+\frac{4}{9}$	$\frac{1}{9}+\frac{1}{9}$	S_2
$\begin{bmatrix}2\\4\end{bmatrix}$	$\frac{64}{9}+\frac{64}{9}$	$\frac{1}{9}+\frac{25}{9}$	S_2

→ no change in assignment!

- Output:

$$\mathbf{m}_1 = \begin{bmatrix} -2/3 \\ 4/3 \end{bmatrix}$$

$$\mathbf{m}_2 = \begin{bmatrix} 5/3 \\ 7/3 \end{bmatrix}$$

$$S_1 = \left\{ \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

$$S_2 = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \end{bmatrix} \right\}$$

Convergence

Prop: The algorithm converges in finitely many steps.

Proof: We want to minimize the error function

$$E(\mathbf{m}) = \sum_{i=1}^{N} \|\mathbf{x}_i - m(\mathbf{x}_i)\|^2 = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in S_j} \|\mathbf{x}_i - \mathbf{m}_j\|^2$$

1. Assign step (2): If a data point $\mathbf{x}_k$ was misplaced in cluster S_j, there is some cluster S_i so that

$$\|\mathbf{x}_k - \mathbf{m}_j\| \geq \|\mathbf{x}_k - \mathbf{m}_i\|$$

   Once the data point gets re-assigned to the correct cluster S_i, then the error function decreases.

2. Update step (3): For a fixed cluster S_j,

   take derivative $\rightarrow$ $$\frac{\partial E}{\partial \mathbf{m}_j} = \sum_{\mathbf{x}_i \in S_j} -2(\mathbf{x}_i - \mathbf{m}_j) = 0$$

   The error is minimized when for each cluster S_j,

$$\sum_{\mathbf{x}_i \in S_j} (\mathbf{x}_i - \mathbf{m}_j) = 0 \Rightarrow \sum_{\vec{x}_i \in S_j} \vec{x}_i = \sum_{\vec{x}_i \in S_j} \vec{m}_j = |S_j| \, \vec{m}_j$$

$$\mathbf{m}_j = \frac{\sum_{\mathbf{x}_i \in S_j} \mathbf{x}_i}{|S_j|}$$

   The error function is convex, so the local minimum is a global minimum $\rightarrow$ the error function decreases.

3. The data set is finite, so a decrease at each step leads to convergence.

- HW 5 due tonight
- Regression project due next Mon
- HW 6 is due following Monday (11/16)
- K-means project a week after (11/23)
- one more hmk at end. HW 7
- Neural networks project (due a week before end of classes) → teams of 2

→ 11/4 work day in class

$\boxed{\text{HW 4}}$ → redo by next Monday

Pb 1, 3 → augment input with $x_0 = 1$ for all data points.

[PLA, regression < linear / logistic, neural nets]