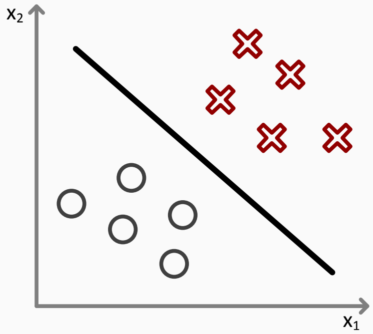
unsupervised learning k-means basics

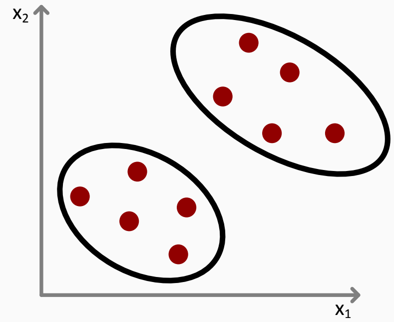
clustering

supervised learning



$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

unsupervised learning

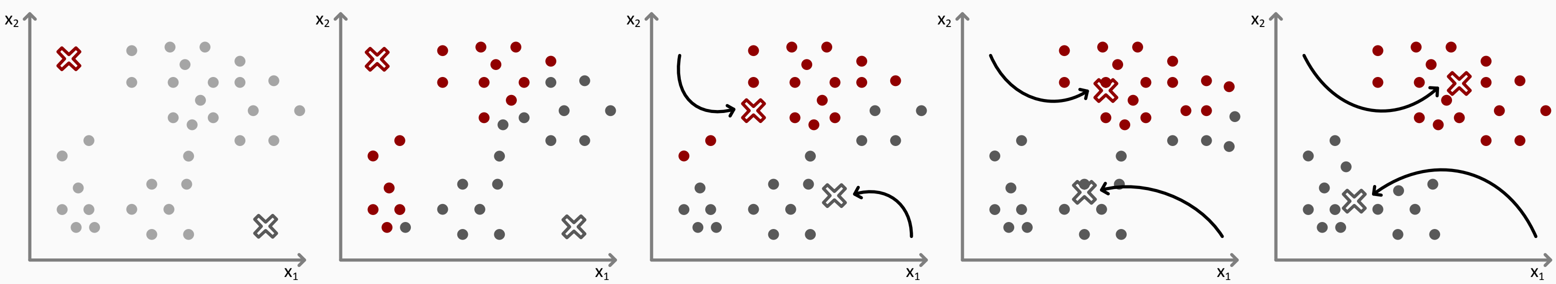


$$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$$

supervised learning algorithms provide training datasets with pairs of known outputs to model off of. unsupervised learning only provides a set of single observations for algorithms like clustering methods are used to find patterns or "structure" in the data

k-means algorithm

iterative algorithm that assigns clusters and adjusts initially randomized cluster centroids to the mean



once the algorithm has converged to the mean, the centroids will no longer adjust or assign color

input:

K (number of clusters)

training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

$x^{(i)} \in \mathbb{R}^n$ (drop $x_0 = 1$ convention $x^{(i)} \in \mathbb{R}^{n+1}$)

randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

repeat {

cluster assignment
- for $i = 1$ to m
  - $c^{(i)} :=$ index (from 1 to K) of cluster centroid closest to $x^{(i)}$

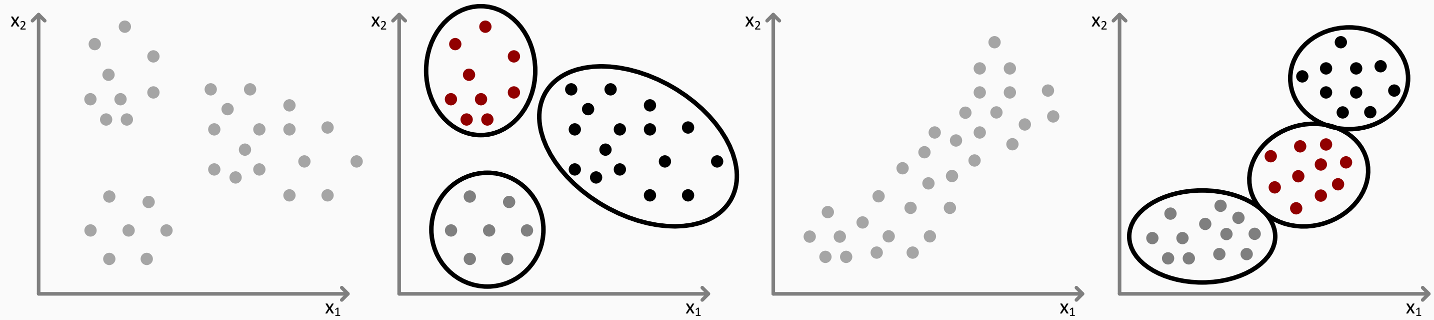
$$c^{(i)} = \min_k \left\| x^{(i)} - u_k \right\|^2$$

move centroid
- for $k = 1$ to K
- $\mu_k :=$ average (mean) of points assigned to cluster k

}

average distance: $\mu_k = \frac{1}{k}\left[x_1^{(i)} + x_2^{(i)} + x_3^{(i)} + \cdots\right] \in \mathbb{R}^n$

k-means for non-separated clusters



optimization objective of k-means

$c^{(i)}$ = index of cluster (1, 2, …, K) to which example $x^{(i)}$ has been assigned

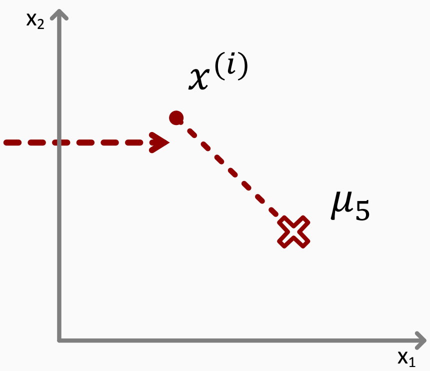
μ_k = cluster of centroid k $\left(\mu_K \in \mathbb{R}^n\right)$

$x^{(i)} \to 5 \quad c^{(i)} = 5 \quad \mu_c{}^{(i)} = \mu_5$

$\mu_c{}^{(i)}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

$$J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K\right) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_c{}^{(i)}\right\|^2$$

distance



$$\min_{\substack{c^{(1)},\dots,c^{(m)},\\ \mu_1,\dots,\mu_K}} J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K\right)$$

cost distortion function

randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

repeat **{**

cluster assignment:
- for $i = 1$ to m
  - $c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

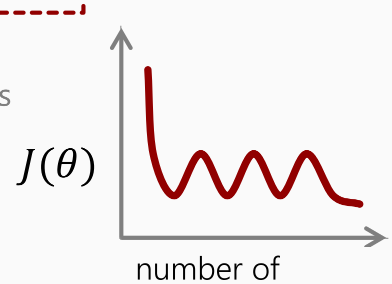
move centroid:
- for $k = 1$ to K
  - μ_k := average (mean) of points assigned to cluster k

}

minimize $J(\dots)$ with respect to the variables $c^{(1)}, c^{(2)}, \dots, c^{(m)}$ while holding $\mu_1, \dots, \mu_K$ fixed

minimize $J(\dots)$ with respect to the variables $\mu_1, \dots, \mu_K$

while running k-means, it is not possible for the cost function to sometimes increase (right illustration); the code should be debugged for errors



random initialization of k-means algorithm

should have $K < m$

$k = 2$

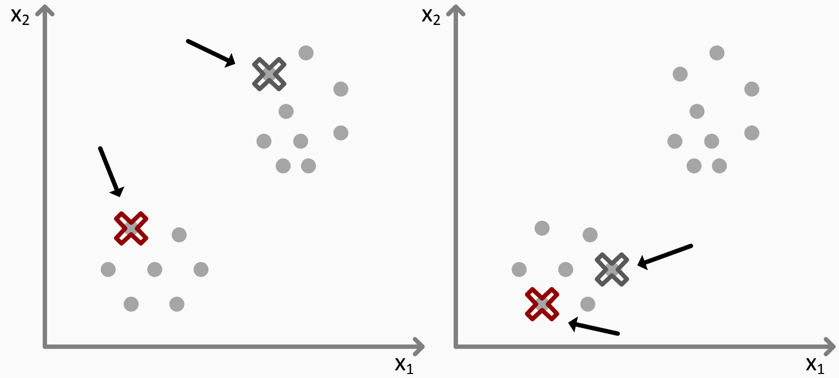
randomly pick K training examples

set $\mu_1, \dots, \mu_K$ equal to the K examples

$\mu_1 = x^{(i)}$

$\mu_1 = x^{(i)}$

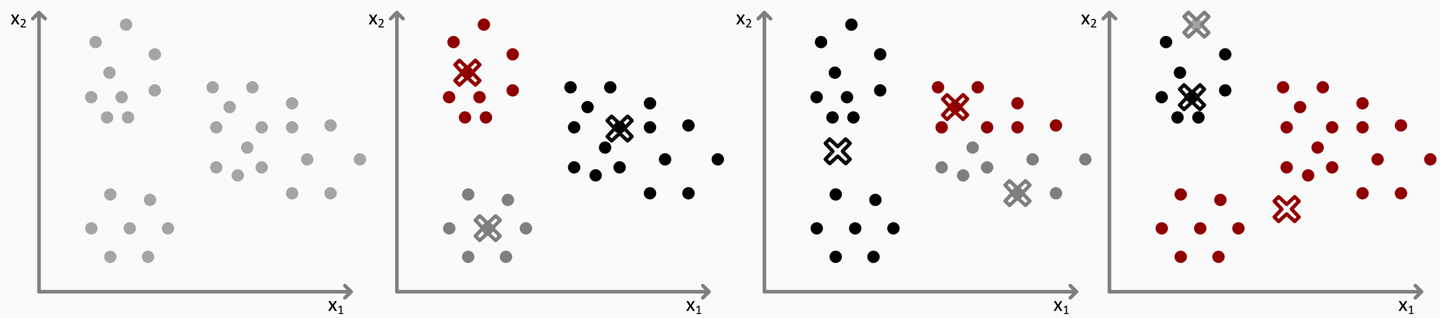
$\vdots$



local optima

k means risks local optima when minimizing the cost distortion function: $J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K\right)$ depending on initial randomization of k

ideal k-means | global optima | local optima | local optima



multiple iterations of random initialization can defend against local optima

for i = 1 to 100 **{**

randomly initialize k-means

run k-means and get $c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K$

compute cost function (distortion)

$$J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K\right)$$

}

the resulting process provides 100 different ways of clustering the data. the clustering method that returns lowest cost $J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K\right)$ will provide the best optima when ***k* is small** (e.g. k = 2-10) then multiple random initializations have a high probability of increasing the effectiveness of clustering optima. it ***k* is large**, there is a better chance the first initial randomization will provide a decent local/global optima.

recommended initialization of k-means:

pick k distinct random integers $i_1, \dots, i_k$, from $\{1, \dots, m\}$

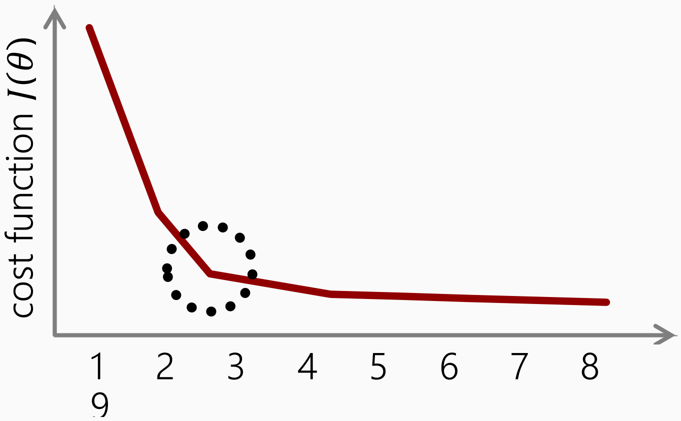
set $u_1 = x^{(i_1)}, u_2 = x^{(i_2)}, \dots, u_k = x^{(i_k)}$

choosing the number of clusters

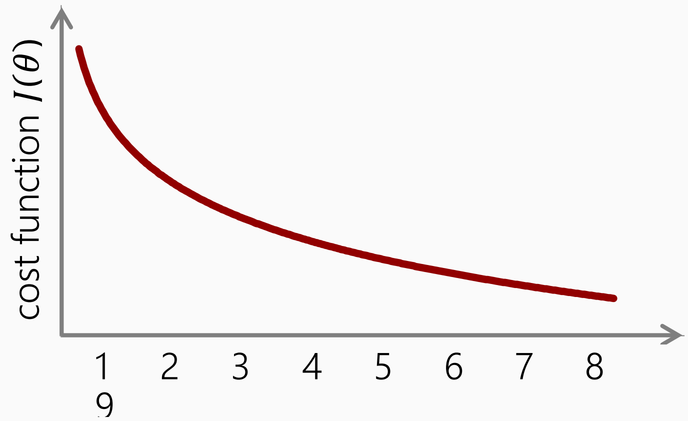
elbow method:

the process of choosing an appropriate amount of k-clusters where the cost function $J(\theta)$ ceases to significantly improve with each additional iteration with new clusters.

in theory



in practice

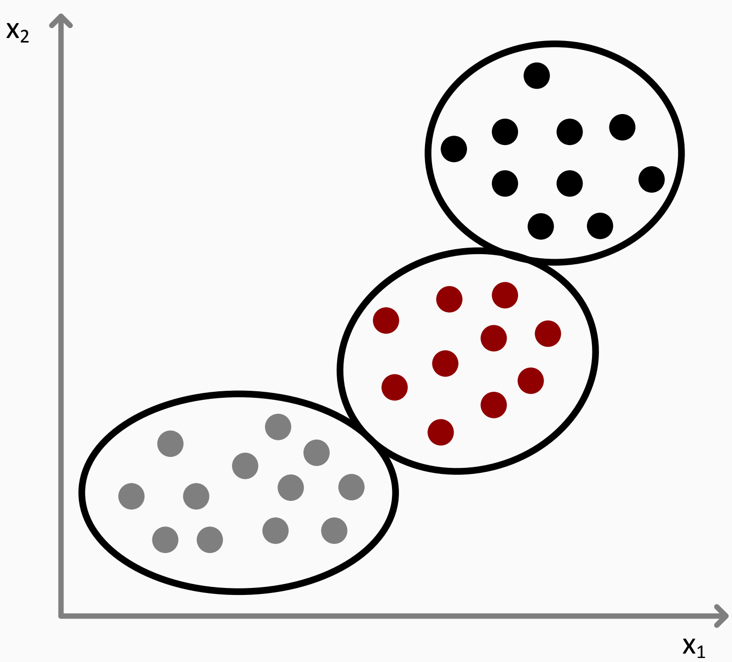


choosing the value of K

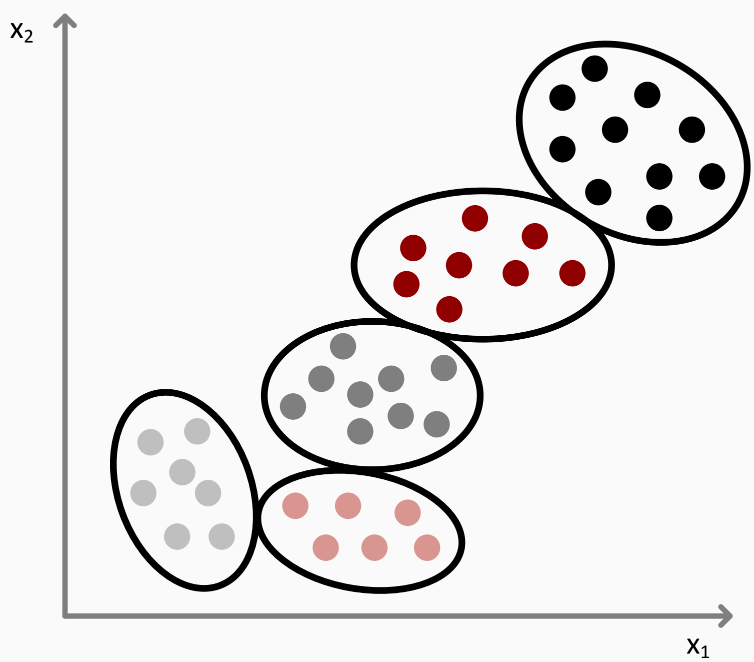
evaluating K-means based on a metric for how well it performs on downstream area

using domain knowledge about the purpose in running k-means can provide insight on how many clusters should be appropriately expected

K = 3



K = 5



if data portrays a certain amount of expected known clusters, K should reflect such