Machine Learning (CSE 446): Unsupervised Learning: The K-means Algorithm

Sham M Kakade

University of Washington
cse446-staff@cs.washington.edu

The Perceptron Convergence Theorem

- Again taking $b = 0$ (absorbing it into w).
- Margin def: Suppose the data are linearly separable, and all data points are γ away from the separating hyperplane. Precisely, there exists a w_*, which we can assume to be of unit norm (without loss of generality), such that for all $(x, y) \in D$.

$$y\,(w_* \cdot x) \geq \gamma$$

γ is the **margin**.

Theorem: (Novikoff, 1962) Suppose the inputs bounded such that $\|x\| \leq R$. Assume our data D is linearly separable with margin γ. Then the perceptron algorithm will make at most $\frac{R^2}{\gamma^2}$ mistakes.
(This implies that at most $O(\frac{N}{\gamma^2})$ updates, after which time w_t never changes.)

Proof of the "Mistake Lemma"

- Let M_t be the number of mistakes at time t.
  If we make a mistake using w_t on (x, y), then observe that $yw_t \cdot x \leq 0$.
- Suppose we make a mistake at time t:

$$w_* \cdot w_t = w_* \cdot (w_{t-1} + yx) = w_* \cdot w_{t-1} + yw_* \cdot x \geq w_* \cdot w_{t-1} + \gamma \,.$$

  Since $w_0 = 0$ and $w_* \cdot w_t$ grows by γ every time we make a mistake, this implies that $w_* \cdot w_t \geq \gamma M_t$.
- Also, if we make a mistake at time t (using that $yw_t \cdot x \leq 0$),

$$\|w_t\|^2 = \|w_{t-1}\|^2 + 2yw_{t-1} \cdot x + ||x||^2 \leq \|w_{t-1}\|^2 + 0 + ||x||^2 \leq \|w_{t-1}\|^2 + R^2 \,.$$

  Since $\|w_t\|^2$ grows by R^2 on every mistake, this implies $\|w_t\|^2 \leq R^2 M_t$ and so $\|w_t\| \leq R\sqrt{M_t}$.
- Now we have that:

$$\gamma M_t \leq w_* \cdot w_t \leq \|w_*\|\|w_t\| \leq R\sqrt{M_t} \,.$$

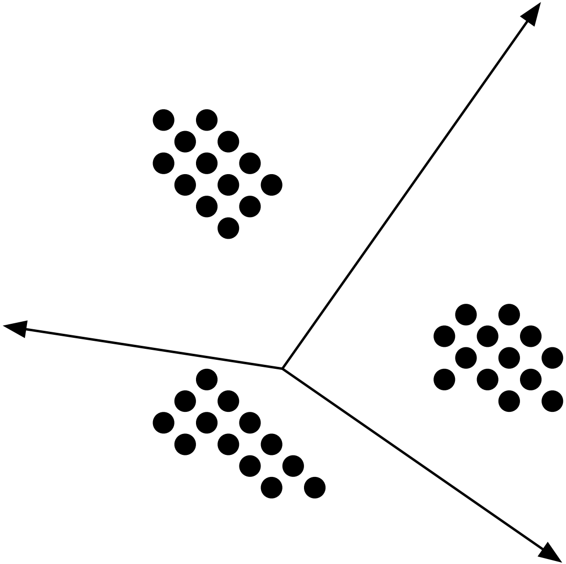
  solving for M_t completes the proof.

Today: Unsupervised Learning and the K-means algorithm

- The Our dataset consists only of inputs: $\{x_1, \ldots x_N\}$.
  Suppose we do not have labels.
- Simple objective: cluster into K groups.

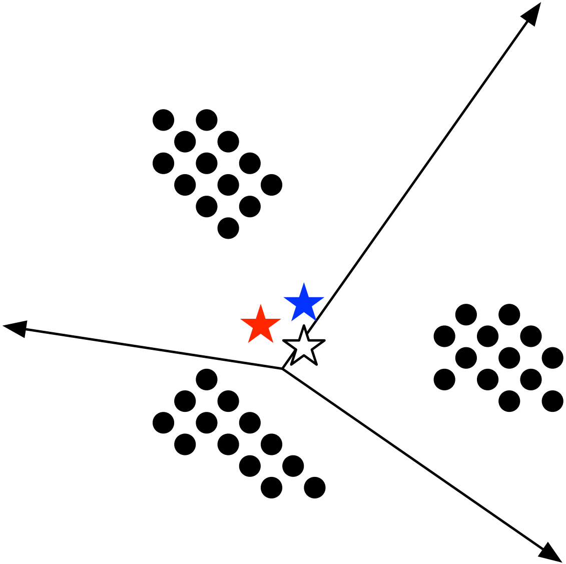
K-Means: An Iterative Clustering Algorithm

(Review from last week.)



K-Means: An Iterative Clustering Algorithm

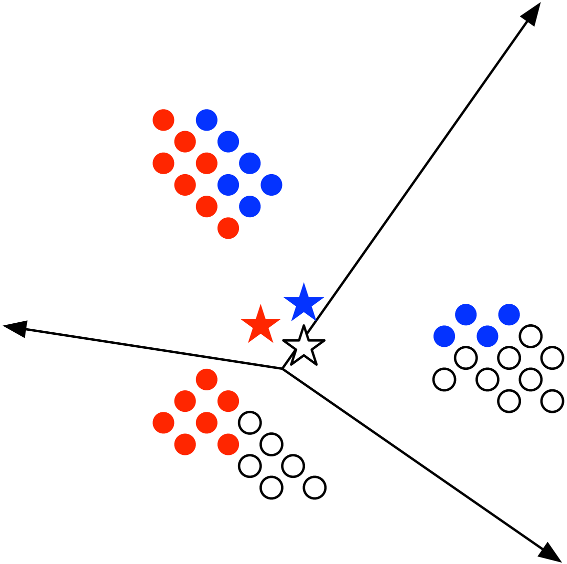
(Review from last week.)



The stars are **cluster centers**, randomly assigned at first.

K-Means: An Iterative Clustering Algorithm

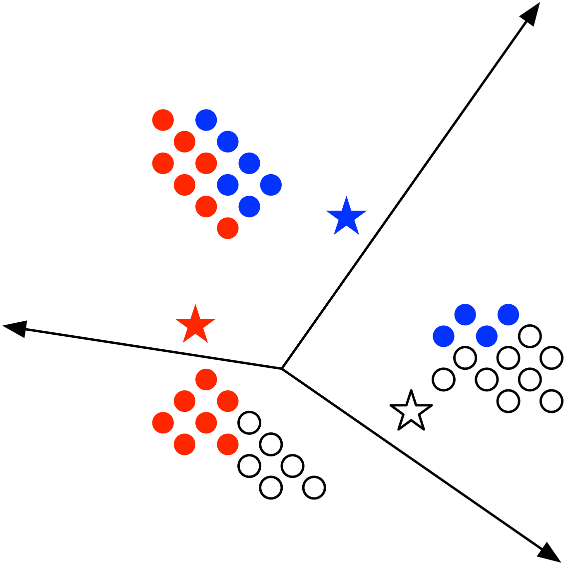
(Review from last week.)



Assign each example to its nearest cluster center.

K-Means: An Iterative Clustering Algorithm

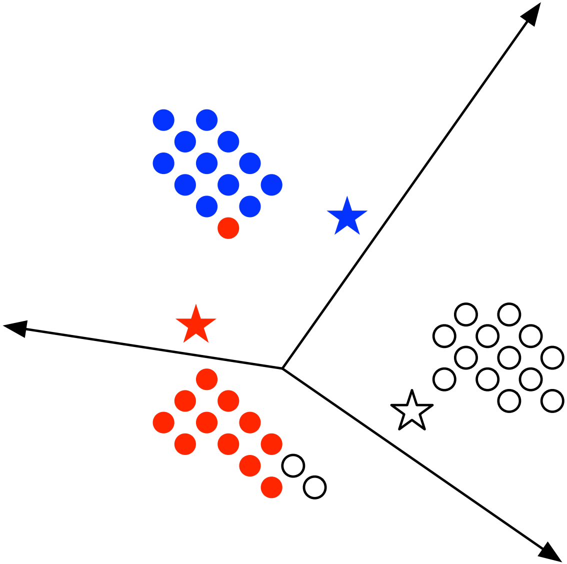
(Review from last week.)



Recalculate cluster centers to reflect their respective examples.

K-Means: An Iterative Clustering Algorithm

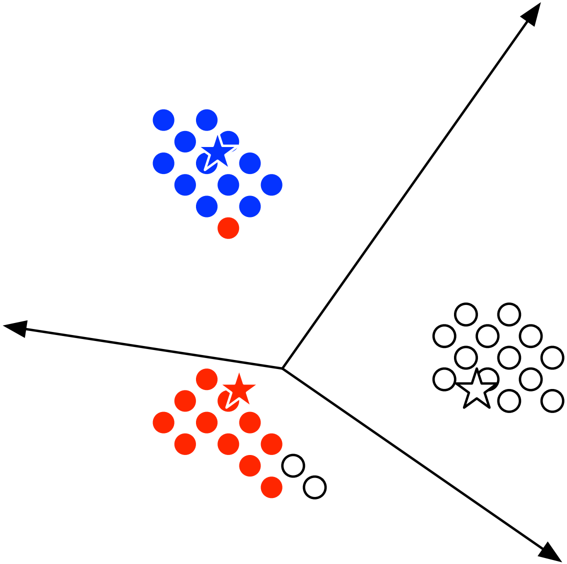
(Review from last week.)



Assign each example to its nearest cluster center.

K-Means: An Iterative Clustering Algorithm

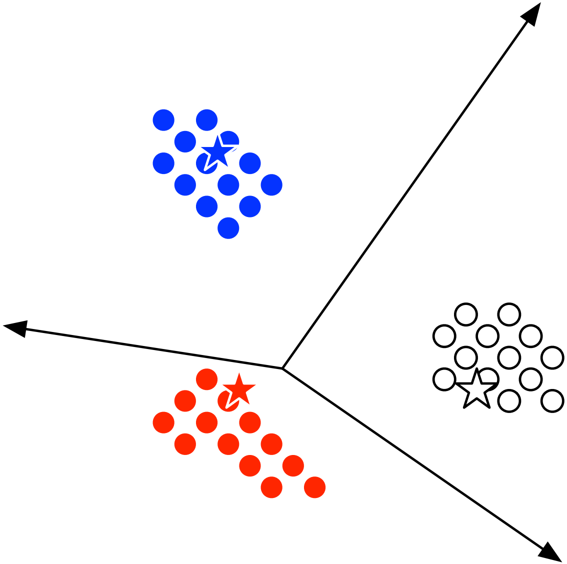
(Review from last week.)



Recalculate cluster centers to reflect their respective examples.

K-Means: An Iterative Clustering Algorithm

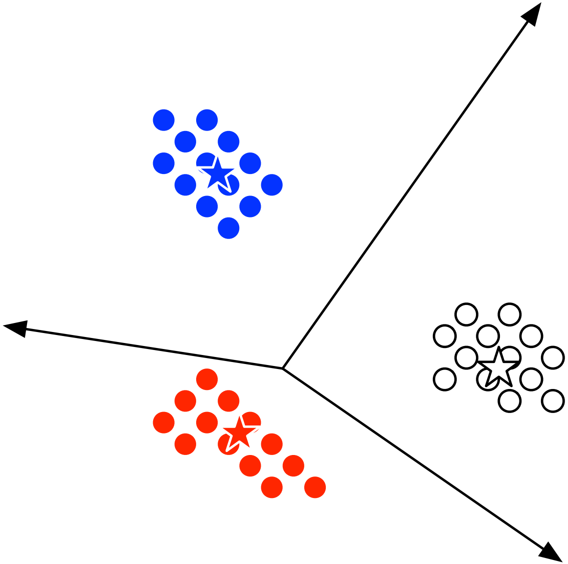
(Review from last week.)



Assign each example to its nearest cluster center.

K-Means: An Iterative Clustering Algorithm

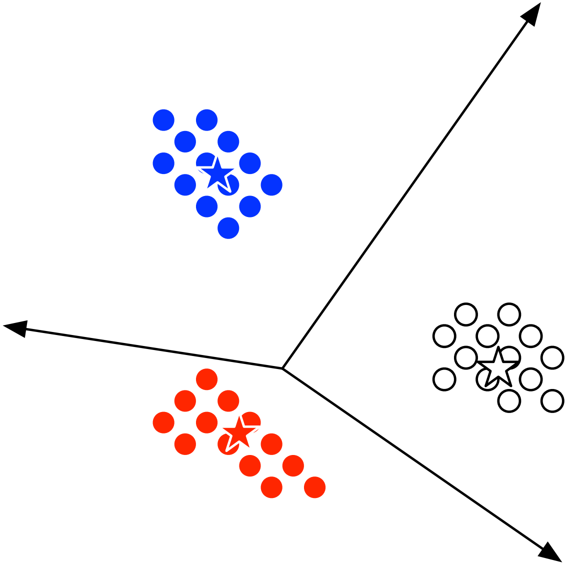
(Review from last week.)



Recalculate cluster centers to reflect their respective examples.

K-Means: An Iterative Clustering Algorithm

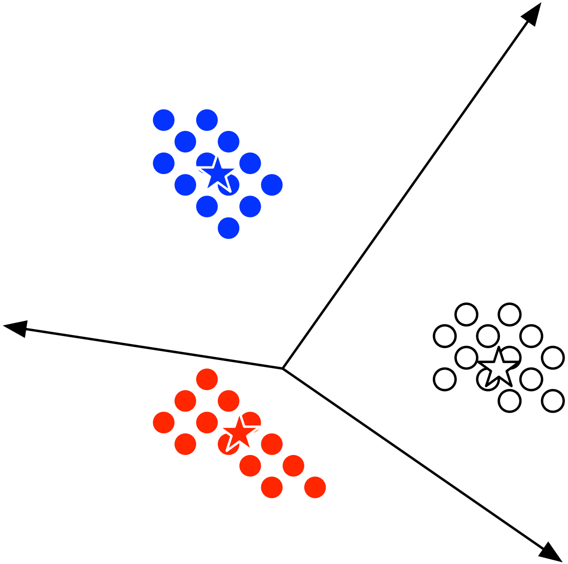
(Review from last week.)



At this point, nothing will change; we have converged.

K-Means: An Iterative Clustering Algorithm

(Review from last week.)



At this point, nothing will change; we have converged.

1. Does it always converge?
   Yes.
2. Does it converge to the "right" answer?
   Not necessarily.

K-Means Clustering

Data: unlabeled data $D = \langle \mathbf{x}_n \rangle_{n=1}^{N}$, number of clusters K
Result: cluster assignment z_n for each $\mathbf{x}_n$
initialize each $\boldsymbol{\mu}_k$ to a random location, for $k \in \{1, \ldots, K\}$;
do
    for $n \in \{1, \ldots, N\}$ **do**
        # assign each data point to its nearest cluster-center let
        $z_n = \operatorname{argmin}_k \|\boldsymbol{\mu}_k - \mathbf{x}_n\|$;
    end
    for $k \in \{1, \ldots, K\}$ **do**
        # recenter each cluster
        let $\boldsymbol{X}_k = \{\mathbf{x}_n \mid z_n = k\}$;
        let $\boldsymbol{\mu}_k = \text{mean}(\boldsymbol{X}_k)$;
    end
while *any* z_n *changes from previous iteration*;
return $\{z_n\}_{n=1}^{N}$;

Algorithm 1: K-MEANS

What would we like to do?

- **Objective function:** find k-means, $\mu_1, \ldots \mu_k$, which minimizes the following squared distance cost function:

$$\sum_{n=1}^{N} \left(\min_{k' \in \{1,\ldots,k-1\}} \|\mathbf{x}_n - \boldsymbol{\mu}_{k'}\|^2 \right)$$

- We can also write this objective function in terms of the assignments z_n's. How?

This is the general approach of loss function minimization: find parameters which make our objection function "small" (and which also "generalizes")

Convergence Proof Sketch

- The cluster assignments, the z_n's take only finitely many values. So the cluster centers, the $\boldsymbol{\mu}_k$'s, also must only take a finite number of values. Each time we update any of them, we will never increase this function:

$$L(z_1, \ldots, z_N, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) = \sum_{n=1}^{N} \left\| \mathbf{x}_n - \boldsymbol{\mu}_{z_n} \right\|^2 \geq 0$$

  L is the **objective function** of K-Means clustering.
- Convergence must occur in a **finite number** of steps, due to:
  L decreases at every step; L can only take on finitely many values.
  See CIML, Chapter 15 for more details.
- Does the solution depend on the random initialization of the means $\boldsymbol{\mu}_*$?

Does K-means converge to the minimal cost solution?

- No! The objective is an NP-Hard problem, so we can't expect **any** algorithm to minimize the cost without essentially checking (near to) all assignments.
- Bad example for K-means:

References I

A. B. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, 1962.