Data Mining & Machine Learning

CS37300
Purdue University

Nov 1, 2023

Optimization of NNs

- An accelerated variant of Batch SGD
  - Can parallelize gradient evaluations across multi-core architectures
  - Batch size has a regularizing effect
- Adaptive learning rate
- Challenges
  - Local minima, saddle points
  - Vanishing/Exploding gradients
  - Ill-conditioned Hessians

Today's topics

Unsupervised learning

- Descriptive modeling: representation
- Partition-based clustering
    - k-means

Descriptive models

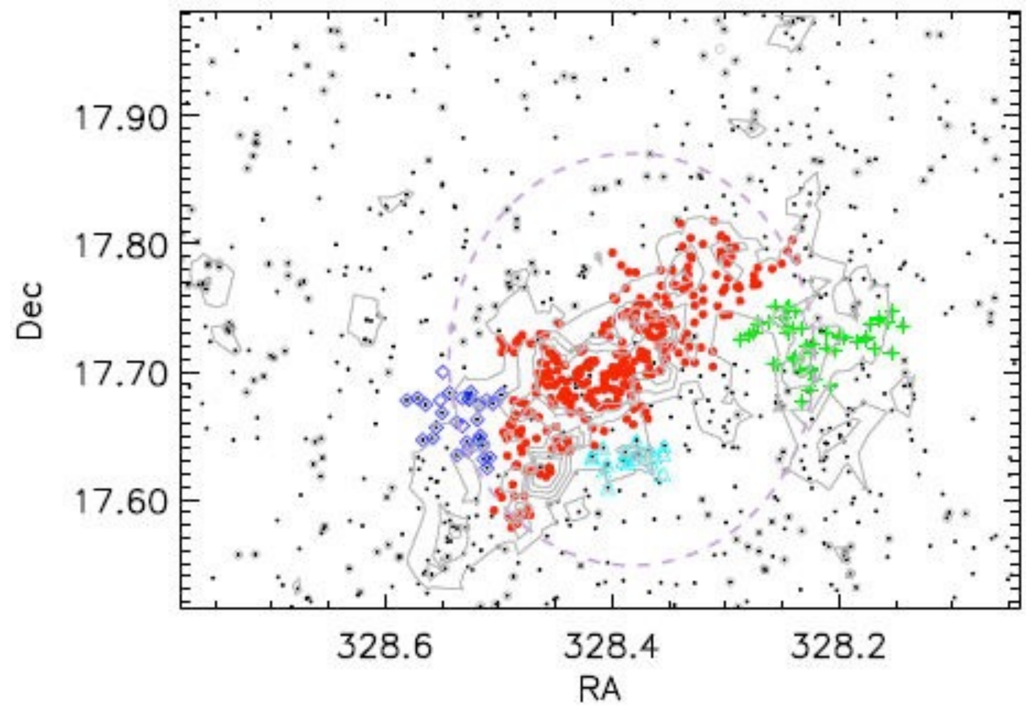
- Descriptive models summarize the data
  - Global summary
  - Model main features of the data
- Two main approaches:
  - Cluster analysis
  - Density estimation

Modeling task

- Data representation: training set $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n$ where each $\underline{x}_i \in R^d$
- Task—depends on approach
  - Clustering: partition the instances into groups of similar instances
  - Density estimation: determine a compact representation of the full joint distribution of the random variable $\underline{X} \in R^d$, that is, $P(\underline{X}) = P(X_1, X_2, \ldots, X_d)$

Cluster analysis

- Decompose or partition samples into groups such that:
    - **Intra**-group similarity is ***high***
    - **Inter**-group similarity is ***low***
- Measure of distance/similarity is crucial



Cluster analysis

- Huge body of work
  - Also known as unsupervised learning, segmentation, etc.
- Difficult to evaluate success
  - If goal is to find “interesting” clusters, then it is difficult to quantify
  - If goal is to find “similar” clusters, then success depends on distance measure

Application examples

- **Marketing**: discover distinct groups in customer base to develop targeted marketing programs
- **Land use**: identify areas of similar use in an earth observation database to understand geographic similarities
- **City-planning**: group houses according to house type, value, and location to identify "neighborhoods"
- **Earth-quake studies**: Group observed earthquakes to see if they cluster along continent faults

Clustering algorithms

- Types:
  - Partition-based methods
  - Hierarchical clustering (divisive/agglomerative)
  - Probabilistic model-based methods
- Different algorithms find clusters of different “shapes”
  - Appropriate shape will depend on application

Algorithm examples

- k-means clustering (partition-based)
- Spectral clustering (hierarchical-divisive)
- Nearest neighbor clustering (hierarchical-agglomerative)
- Mixture models (probabilistic model-based)

Today's topics

- Descriptive modeling: representation
- Partition-based clustering
  - k-means

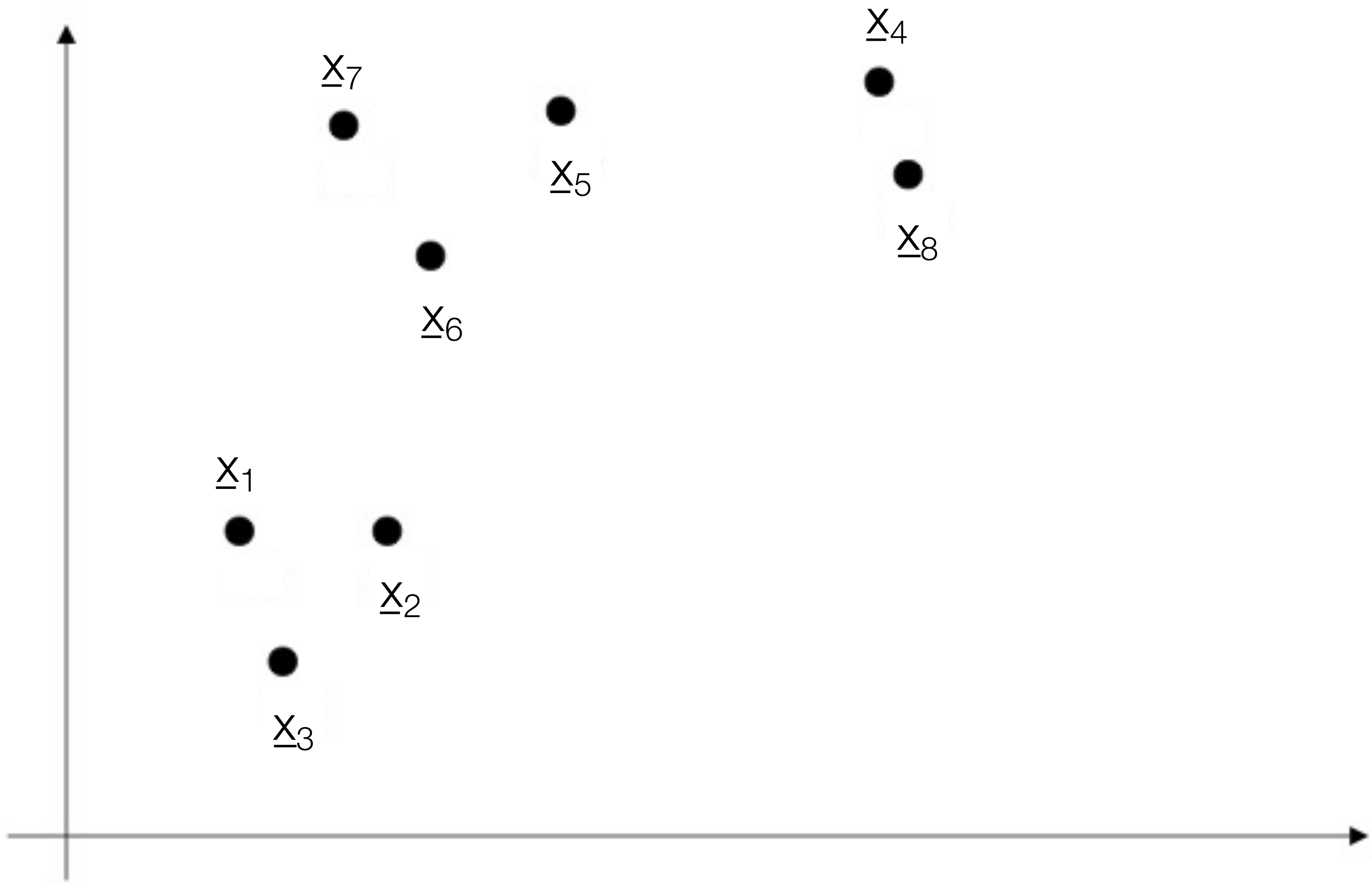
Partition-based

- Input: dataset D = $\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n\}$
- Output: k clusters $C = \{C_1, \ldots, C_k\}$ such that each $\underline{x}_i$ is assigned to a unique C_j
- Evaluation: Score(C,D) is maximized/minimized
  - Combinatorial optimization: search among k^n allocations of n objects into k classes to maximize score function
  - Exhaustive search is intractable
  - Most approaches use iterative improvement algorithms

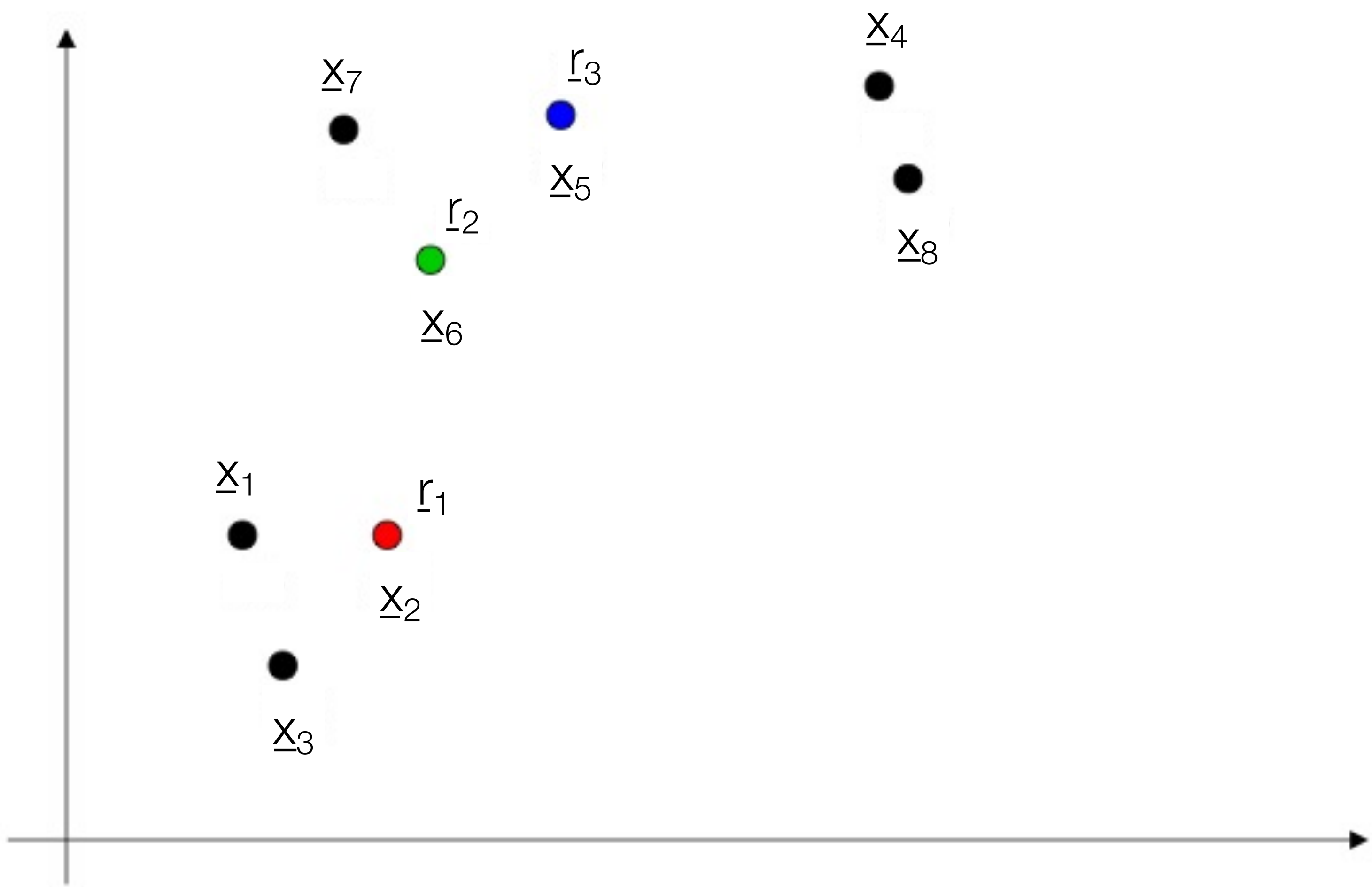
k-means

- Algorithm:
  - Start with k randomly chosen centroids
  - Repeat until no changes in assignments
    - Assign each sample to closest centroid
    - Recompute cluster centroids

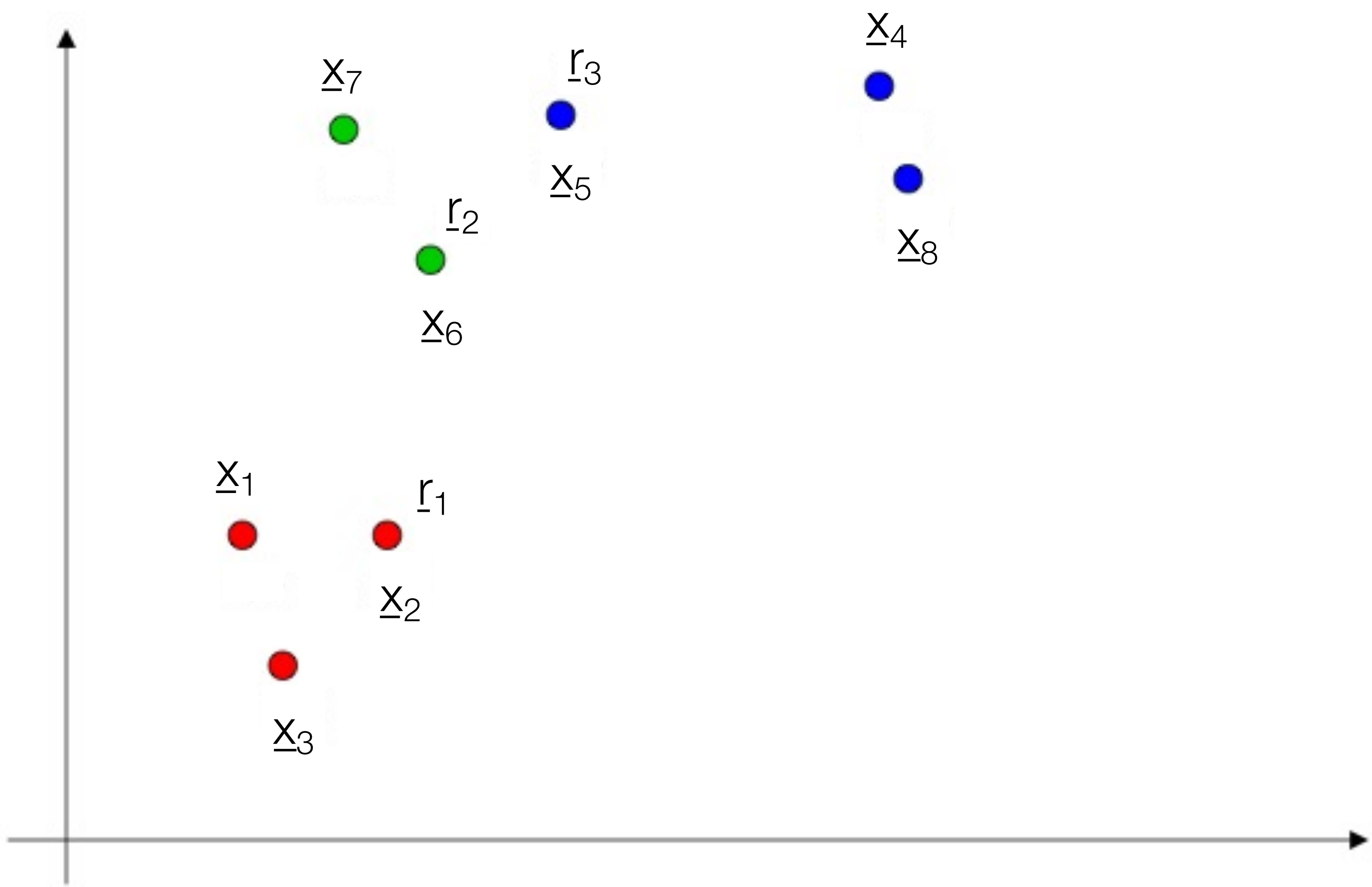
k-means example (here k=3)



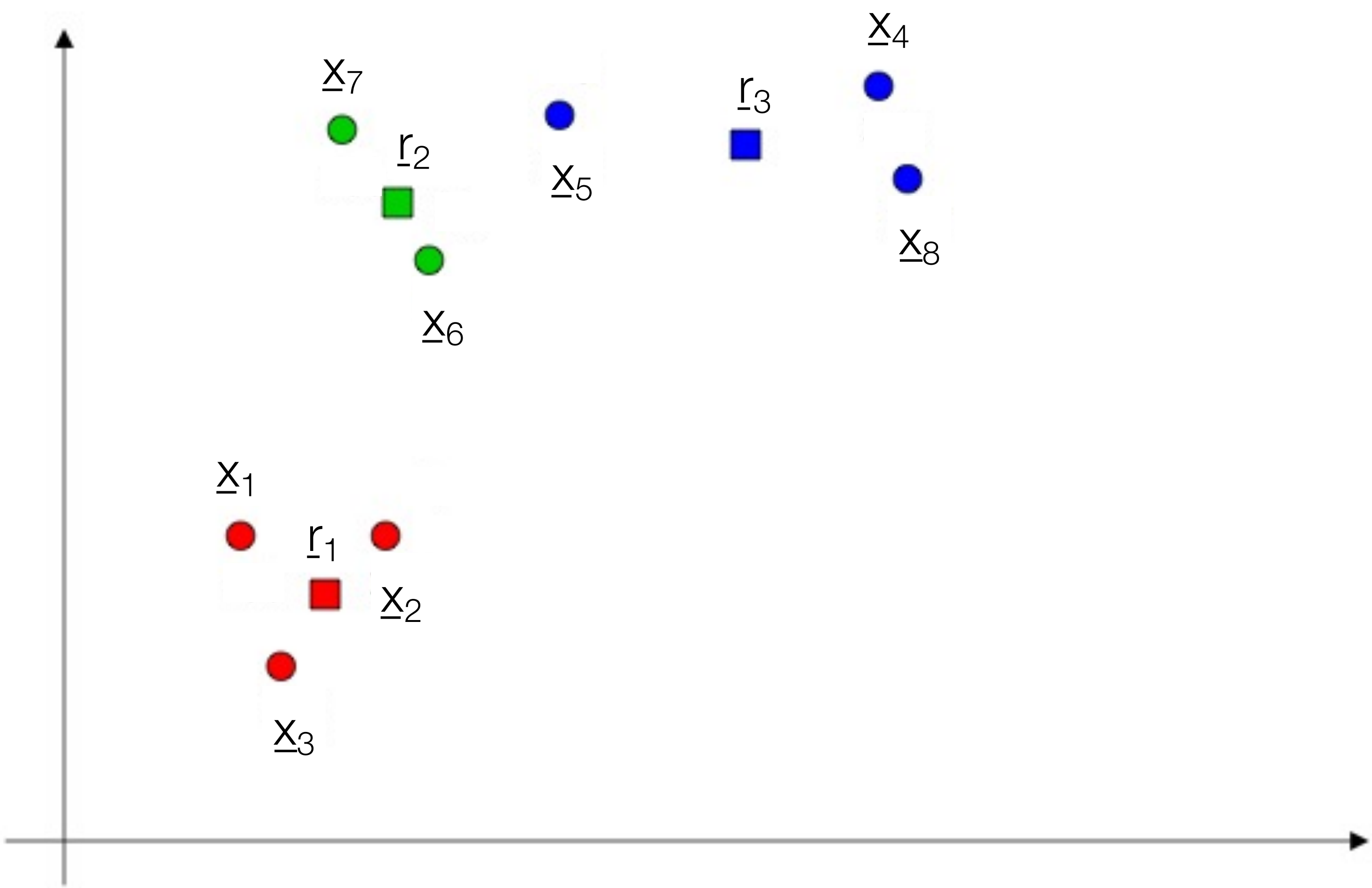
Dataset D = $\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_8\}$



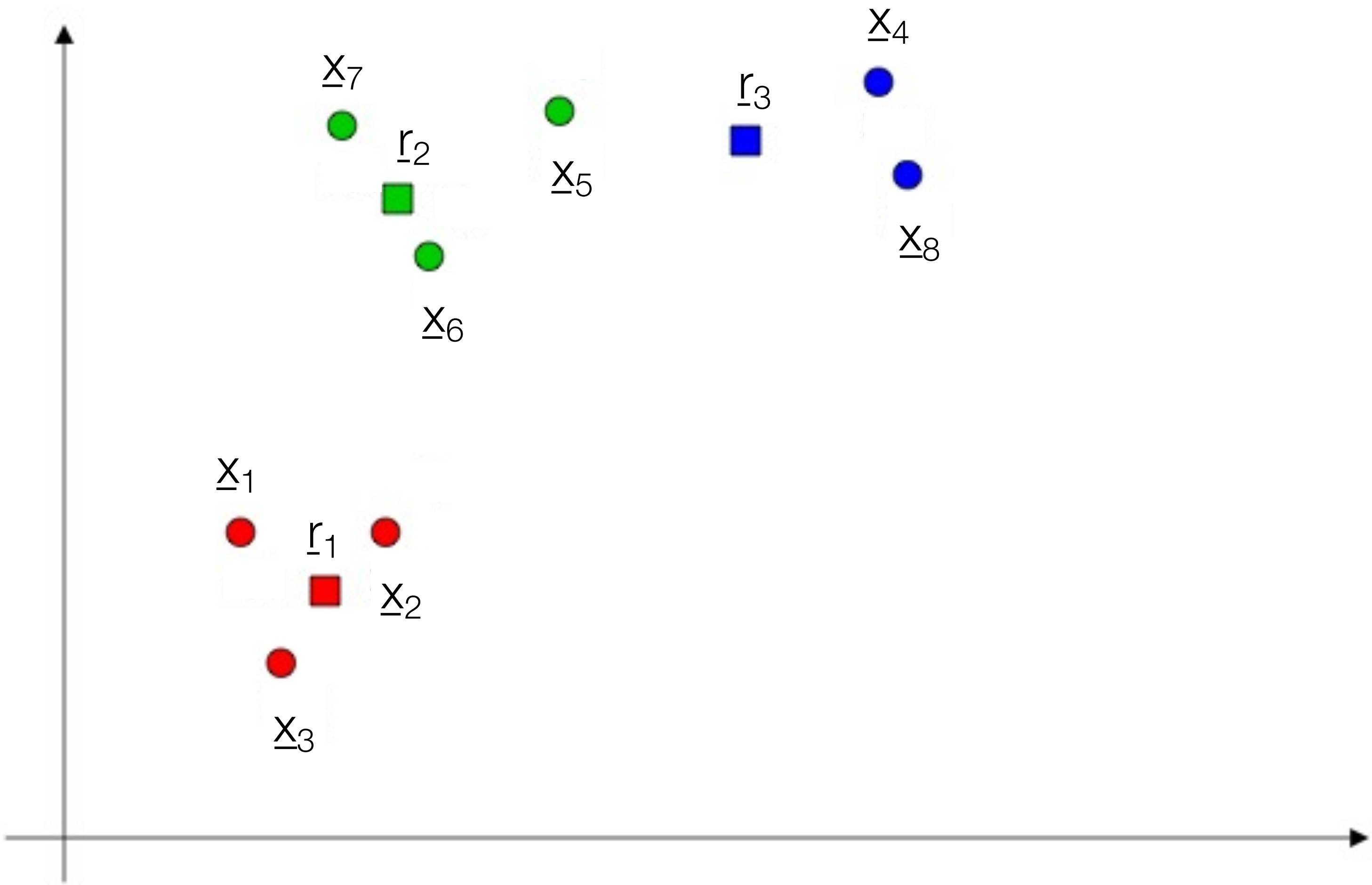
Start with k randomly chosen centroids $\underline{r}_1, \ldots, \underline{r}_k$



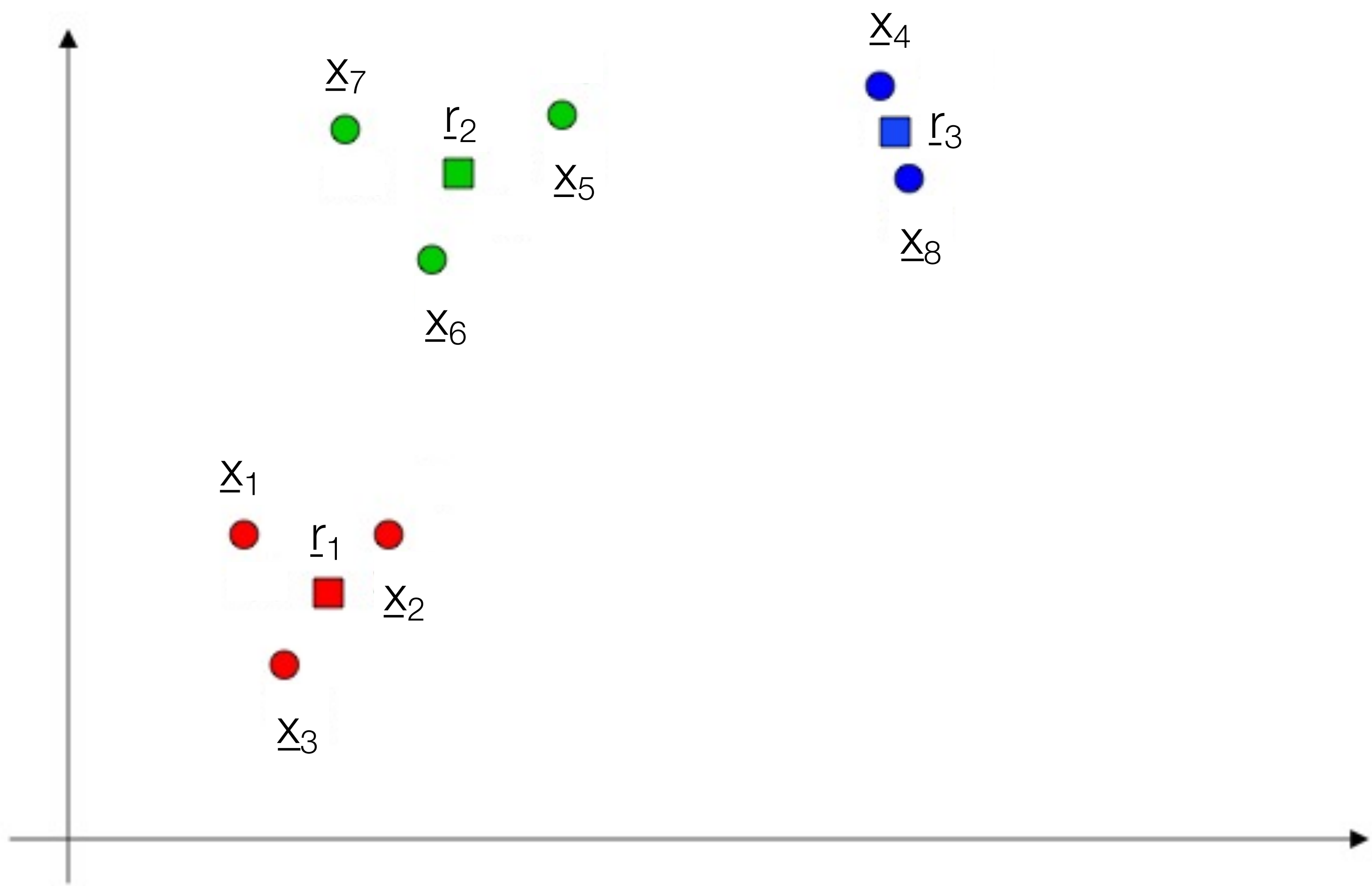
Assign each sample to closest centroid



Recompute cluster centroids $\underline{r}_1, \ldots, \underline{r}_k$



Assign each sample to closest centroid



Recompute cluster centroids $\underline{r}_1, \ldots, \underline{r}_k$

Clustering score functions

- Goal:
  - Compact clusters: minimize within cluster distance
  - Separated clusters: maximize between cluster distance
- score(C,D) = f(wc(C,D), bc(C))
  - Score measures quality of clustering C for dataset D
  - *Many score functions are a combination of within-cluster (wc) and between-cluster (bc) distance measures*

Clustering score functions

- score(C,D) = f(wc(C,D), bc(C))

cluster j centroid:

$$\underline{r}_j = \frac{1}{|C_j|} \sum_{\underline{x}_i \in C_j} \underline{x}_i$$

between-cluster distance:

$$\mathrm{bc}(C) = \sum_{1 \le j \le m \le k} \left\| \underline{r}_j - \underline{r}_m \right\|^2$$

within-cluster distance:

$$\mathrm{wc}(C,D) = \sum_{j=1}^{k} \sum_{\underline{x}_i \in C_j} \left\| \underline{x}_i - \underline{r}_j \right\|^2$$

k-means

- Algorithm:
  - Start with k randomly chosen centroids
  - Repeat until no changes in assignments
    - Assign each sample to closest centroid
    - Recompute cluster centroids (average of points in the cluster)

Score function: $$\mathrm{score}(C,D) = \mathrm{wc}(C,D) = \sum_{j=1}^{k} \sum_{\underline{x}_i \in C_j} \left\| \underline{x}_i - \underline{r}_j \right\|^2$$

k-means

- Does it terminate?
  - Yes, the objective function decreases on each iteration. It usually converges quickly.
- Does it converge to an optimal solution?
  - No, the algorithm terminates at a local optima which depends on the starting seeds.
- What is the time complexity?
  - $\propto$ k n L , where L is the number of iterations

k-means

- Strengths:
  - Relatively efficient
  - Finds spherical clusters
- Weaknesses:
  - Terminates at local optimum (sensitive to initial seeds)
  - Applicable only when mean is defined
  - Susceptible to outliers/noise

Variations

- Selection of initial centroids
  - Run with multiple random selections, pick result with best score
  - Use hierarchical clustering to identify likely clusters and pick seeds from distinct groups
- Algorithm modifications:
  - Recompute centroid after each point is assigned
  - Allow for merge and split of clusters (for instance, if cluster becomes empty, start a new one from randomly selected point)

K-means++

- Selection of initial centroids
  - Choose a first center uniformly at random from the data points
  - For each data point x, computer D(x)= distance from x to the nearest center that has already been chosen
  - Choose the next center randomly according to a probability distribution P with P(x) proportional to $D(x)^2$ for each x
  - Repeat until we have k centers chosen, then run the k-means algorithm

- With this initialization, k-means typically converges faster
- Also, this initialization guarantees (in expectation) that the k-means score function upon convergence is withing a factor log(k) of optimal

k-means summary

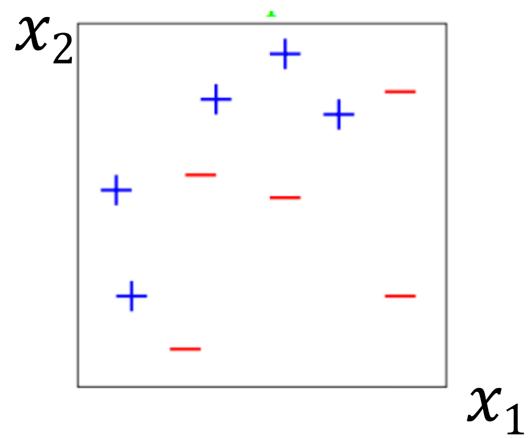
- Knowledge representation
  - k clusters are defined by canonical members (centroids)
- Model space the algorithm searches over?
  - *All possible partitions of the examples into k groups*
- Score function?
  - *Minimize within-cluster Euclidean distance*
- Search procedure?
  - *Iterative refinement correspond to greedy hill-climbing*

Take-home Quiz

Due Nov 3 at 9am on Gradescope

- Please build an AdaBoost ensemble of decision stumps on the following data. Make sure to include detailed steps in your submission.

x1	x2	Decision
2	3	true
2.1	2	true
4.5	6	true
4	3.5	false
3.5	1	false
5	7	true
5	3	false
6	5.5	true
8	6	false
8	2	false



Visualization of the data

Hint: The correct ensemble only uses three decision stumps to achieve 100% accuracy.