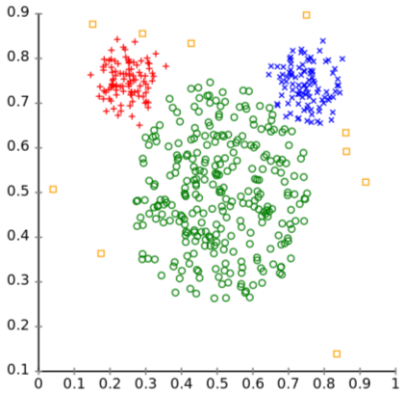
CLUSTERING

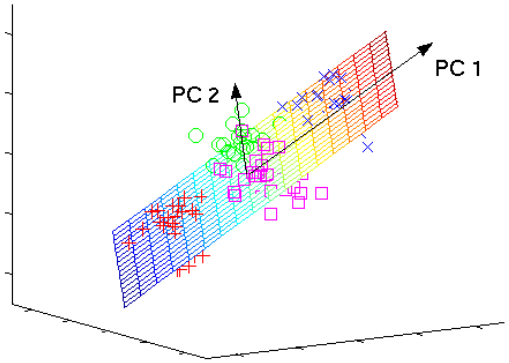
UNSUPERVISED LEARNING

- No labels/responses. Finding structure in data.
- Dimensionality Reduction.



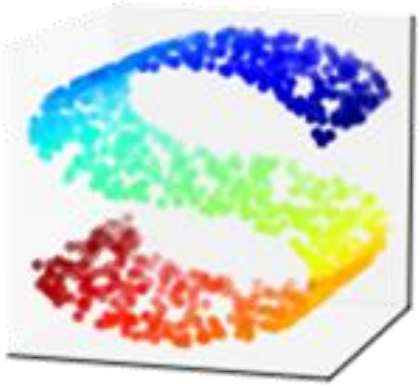
Clustering

$T: \mathbb{R}^d \rightarrow \{1, 2, \ldots, k\}$



Subspace Learning

$T: \mathbb{R}^d \rightarrow \mathbb{R}^m$

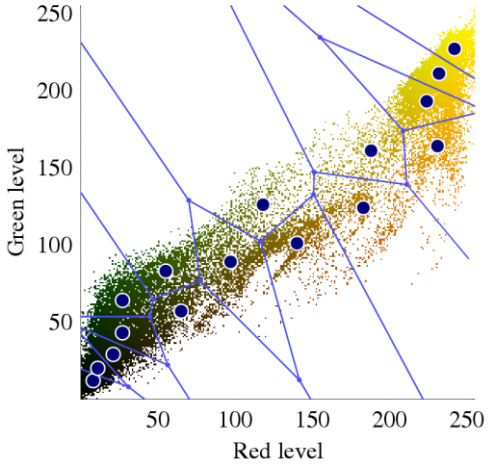


Manifold Learning

USES OF UNSUPERVISED LEARNING

- Data compression

RGB
(12,165,73)
(11,167,79)
:



Labels
3
43
:

Dictionary
1 ~ (10, 160, 70)
2 ~ (40, 240, 20)
:

USES OF UNSUPERVISED LEARNING

- Improve classification/regression (semi-supervised learning)

1. From *unlabeled data,* learn a good features $T: \mathbb{R}^d \to \mathbb{R}^m$.
2. To *labeled data*, apply transformation $T: \mathbb{R}^d \to \mathbb{R}^m$.

$$\left(T\left(x^{(1)}\right), y^{(1)}\right), \ldots, \left(T\left(x^{(n)}\right), y^{(n)}\right)$$

3. Perform classification/regression on transformed data.

ROADMAP

Clustering

- K-Means Algorithm

Dim Reduction with Complete Features

Unsupervised

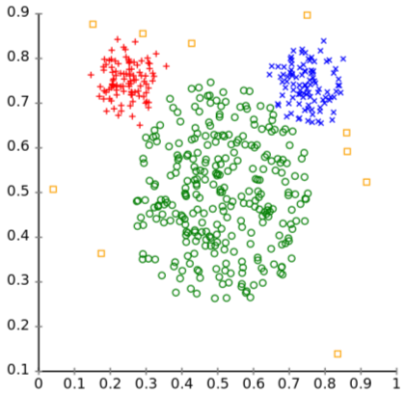
Recommender Systems
Collaborative Filtering
Missing Data Prediction

- K-Nearest Neighbors
- Matrix Factorization

Dim Reduction with Incomplete Features

Supervised or Unsupervised?

WHAT IS CLUSTERING



Clustering Problem.

Input. Training data $S_n = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$, each $x^{(i)} \in \mathbb{R}^d$.
Integer k

Output. Clusters $C_1, C_2, \ldots, C_k \subset \{1, 2, \ldots, n\}$ such that every data point is in one and only one cluster.

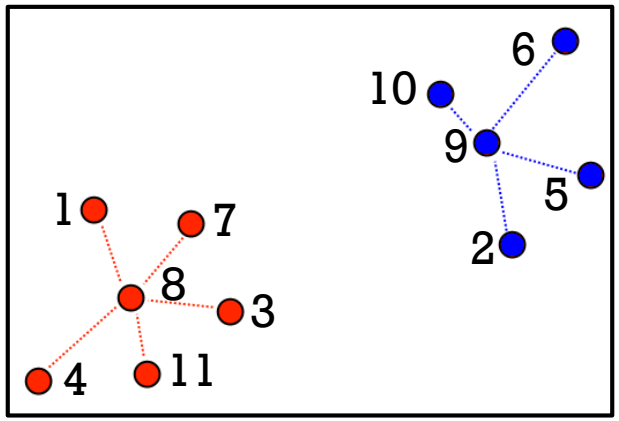
Some clusters could be empty!

HOW TO SPECIFY A CLUSTER

- By listing all its elements

$$\mathcal{C}_1 = \{1,3,4,7,8,11\}$$

$$\mathcal{C}_2 = \{2,5,6,9,10\}$$

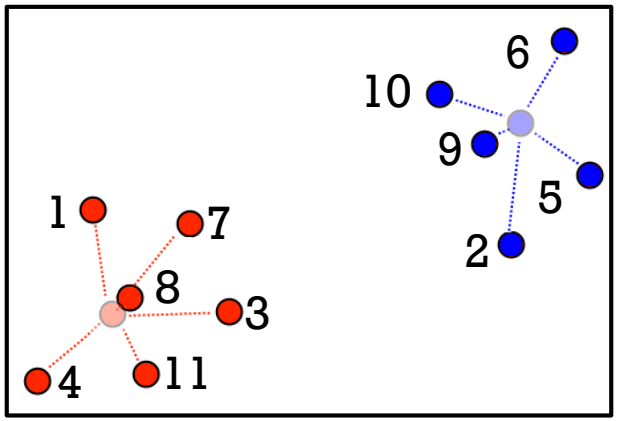


HOW TO SPECIFY A CLUSTER

Each point $x^{(i)}$ will be assigned the closest representative.

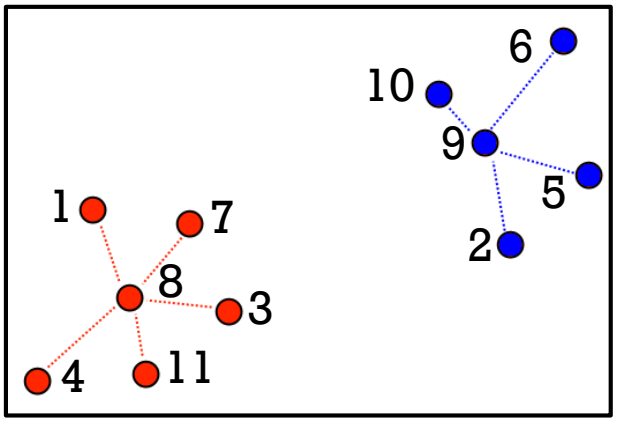
- Using a representative
  a. A point in center of cluster (centroid)
  b. A point in the training data (exemplar)

$$z^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, z^{(2)} = \begin{pmatrix} 5 \\ 4 \end{pmatrix}$$



centroid

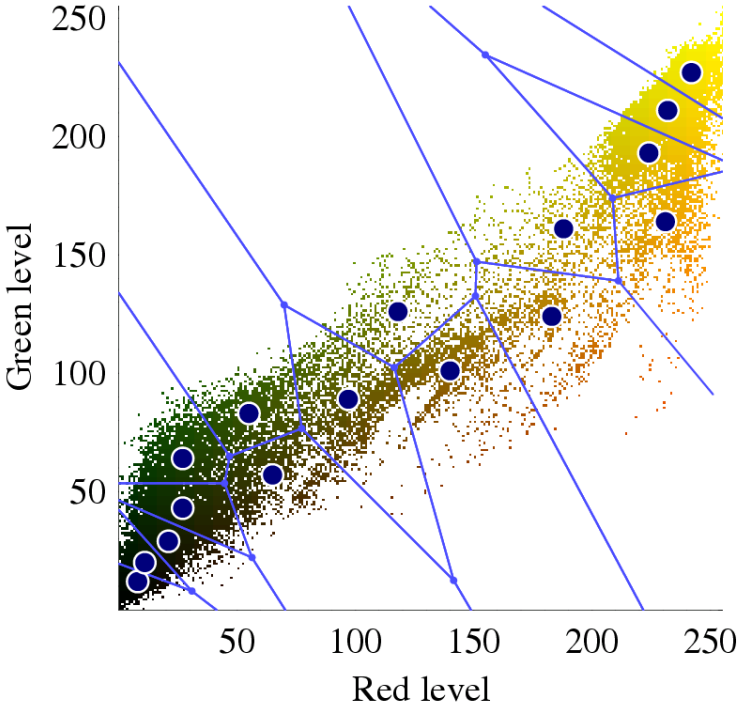
$$z^{(1)} = 8, z^{(2)} = 9$$



exemplar

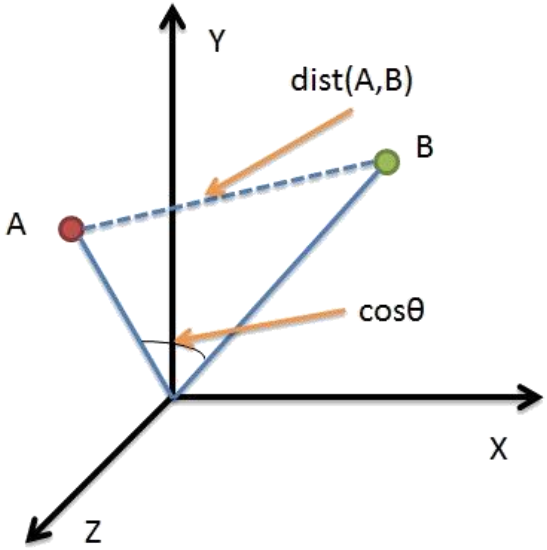
VORONOI DIAGRAM

We can partition all the points in the space into regions, according to their closest representative.



TRAINING LOSS

DISTANCE METRICS

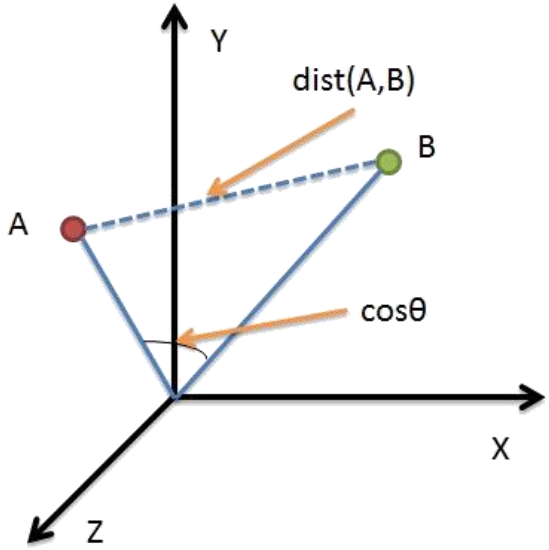


(sometimes called *loss functions*)

A measure of how close two data points are. Nearby points (i.e. distance is *small*) are more likely they belong to the same cluster.

- Euclidean Distance $\text{dist}(x, y) = \|x - y\|^2$

SIMILARITY FUNCTIONS



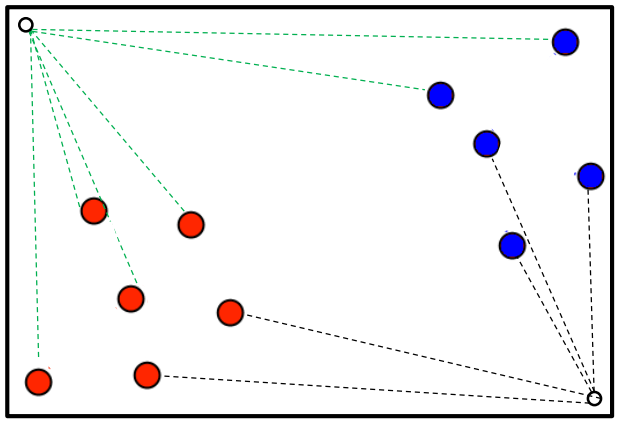
(sometimes called *kernels, correlation*)

A measure of how alike two data points are. Similar points (i.e. similarity is *large*) are more likely they belong to the same cluster.

- Cosine Similarity $\cos(x, y) = \frac{x^\top y}{\|x\| \, \|y\|}$

TRAINING LOSS

Sum of squared distances to closest representative.

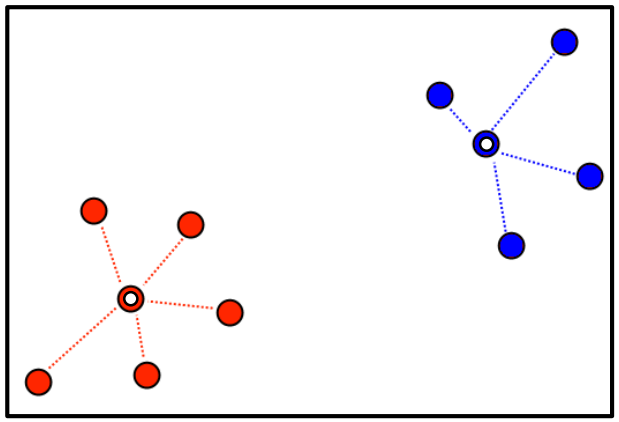


$$\text{loss} \approx 11 \times (1)^2 = 11$$

assume length of each edge is about 1

TRAINING LOSS

Sum of squared distances to closest representative.



$$\text{loss} \approx 9 \times (0.1)^2 = 0.09$$

assume length of each edge is about 0.1

TRAINING LOSS

How do I use a similarity function instead?

Optimizing over representatives.

$$\mathcal{L}_{n,k}\left(z^{(1)}, \ldots, z^{(k)}; \mathcal{S}_n\right) = \sum_{i=1}^{n} \min_{1 \le j \le k} \left\| x^{(i)} - z^{(j)} \right\|^2 .$$

TRAINING LOSS

Optimizing over clusters.

$$\mathcal{L}_{n,k}(\mathcal{C}_1, \ldots, \mathcal{C}_n; \mathcal{S}_n) = \sum_{j=1}^{n} \sum_{i \in \mathcal{C}_j} \left\| x^{(i)} - \frac{1}{|\mathcal{C}_j|} \sum_{i' \in \mathcal{C}_j} x^{(i')} \right\|^2 .$$

TRAINING LOSS

Instead of the distance metric, you can use the *negative* similarity function.

Optimizing both clusters and representatives.

$$\mathcal{L}_{n,k}\left(\mathcal{C}_1, \ldots, \mathcal{C}_k, z^{(1)}, \ldots, z^{(k)}; \mathcal{S}_n\right) = \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \left\| x^{(i)} - z^{(j)} \right\|^2$$

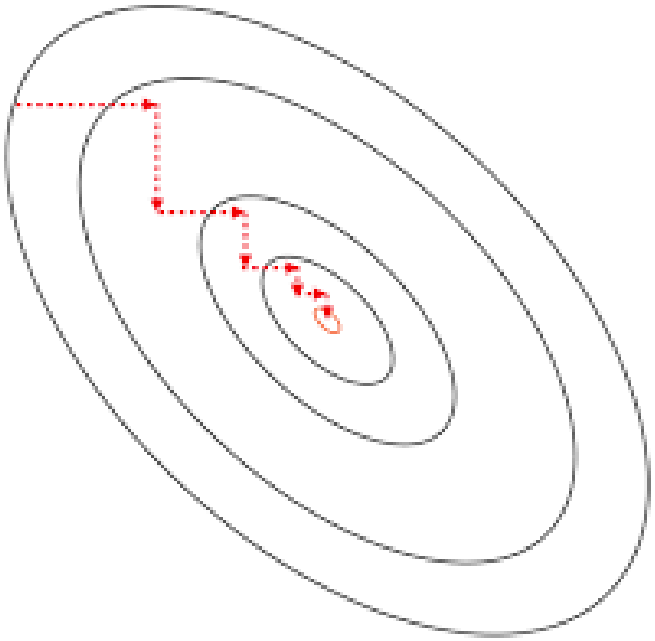
These clusters need not consist of points closest to the representatives.

These representatives need not be the centroids of the clusters.

K-MEANS

OPTIMIZATION ALGORITHM

Goal. Minimize $\mathcal{L}(x, y)$.



Coordinate Descent (Gradient).

Repeat until convergence:
1. Move in direction of $\partial\mathcal{L}/\partial x$.
2. Move in direction of $\partial\mathcal{L}/\partial y$.

Coordinate Descent (Optimization).

Repeat until convergence:
1. Find optimal x while holding y constant.
2. Find optimal y while holding x constant.

OPTIMIZATION ALGORITHM

Coordinate Descent (Optimization)

Repeat until convergence:

- Find best clusters given centroids
- Find best centroid given clusters

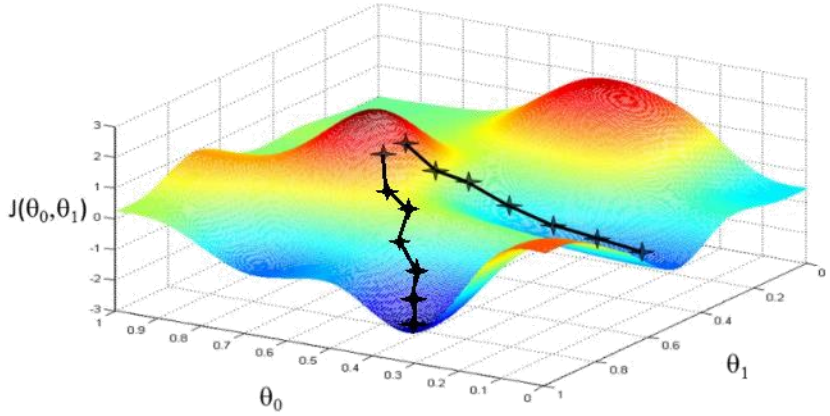
$$\mathcal{L}_{n,k}\left(\mathcal{C}_1, \dots, \mathcal{C}_k, z^{(1)}, \dots, z^{(k)}; \mathcal{S}_n\right) = \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \left\| x^{(i)} - z^{(j)} \right\|^2$$

OPTIMIZATION ALGORITHM

1. Initialize centroids $z^{(1)}, \ldots, z^{(k)}$ from the data.
2. Repeat until no further change in training loss:
   a. For each $j \in \{1, \ldots, k\}$,
      $$\mathcal{C}_j = \{\, i \text{ such that } x^{(i)} \text{ is closest to } z^{(j)} \,\}.$$
   b. For each $j \in \{1, \ldots, k\}$,
      $$z^{(j)} = \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} x^{(i)} \quad \text{(cluster mean)}$$

CONVERGENCE

- Training loss always decreases in each step (coordinate descent).
- Converges to local minimum, not necessarily global minimum.



Challenge.
Why does the algorithm terminate in a finite number of steps?

* not in syllabus

Repeat algorithm over many initial points, and pick the configuration with the smallest training loss.

DISCUSSION

INITIALIZATION

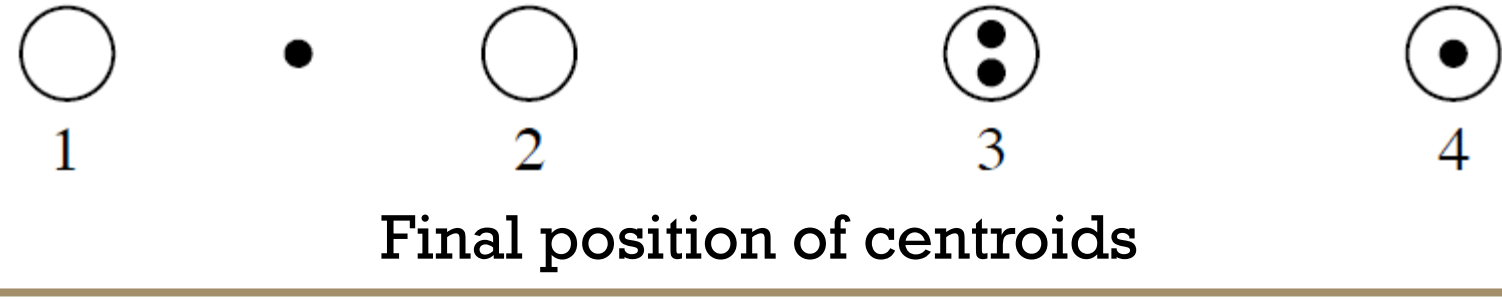
Optimization

- Empty clusters
  - Pick data points to initialize clusters
- Bad local minima
  - Initialize many times and pick solution with smallest training loss
  - Pick good starting positions

INITIALIZATION

Optimization

Starting position of centroids

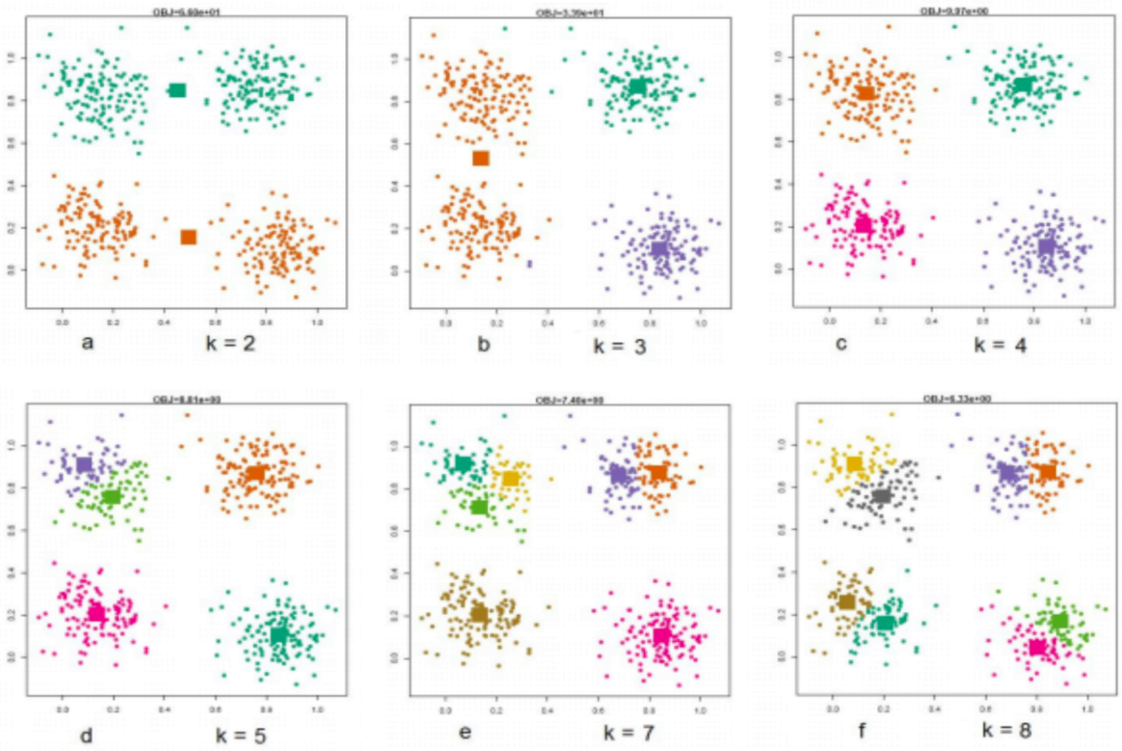


Final position of centroids

Problem. How to choose good starting positions?
Solution. Place them far apart with high probability.

NUMBER OF CLUSTERS

Generalization



NUMBER OF CLUSTERS

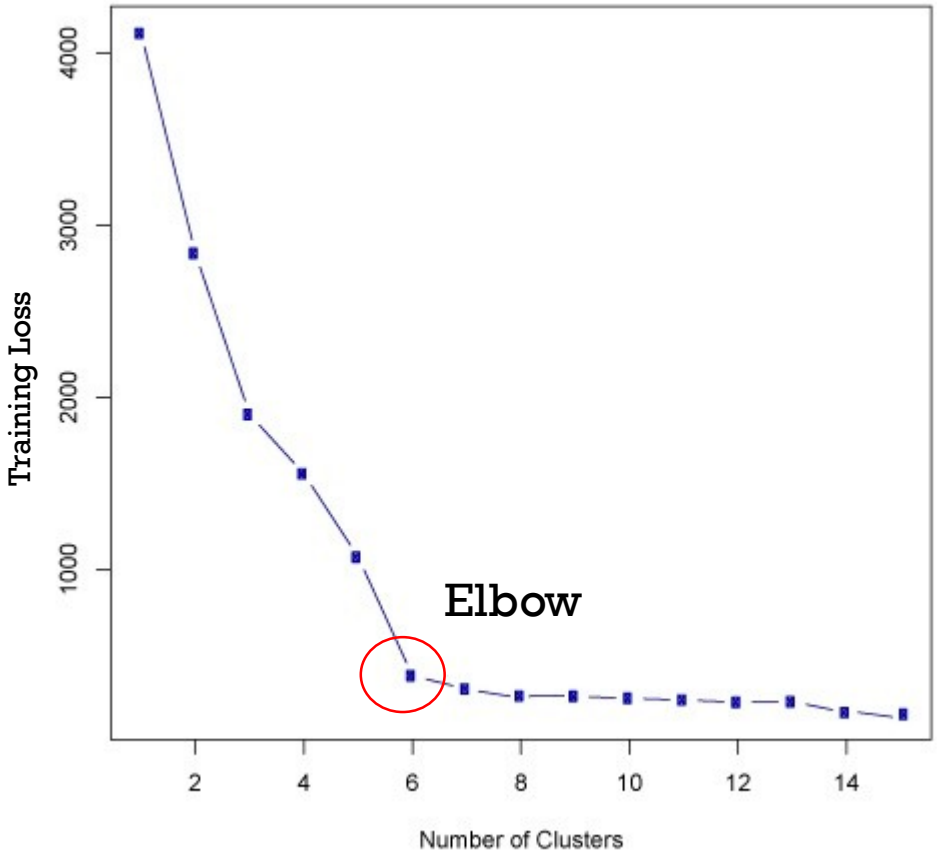
Generalization

How do we choose k, the optimal number of clusters?

- Elbow method
  - Training Loss
  - Validation Loss
- Semi-supervised learning
  - Accuracy in supervised task

ELBOW METHOD

Generalization

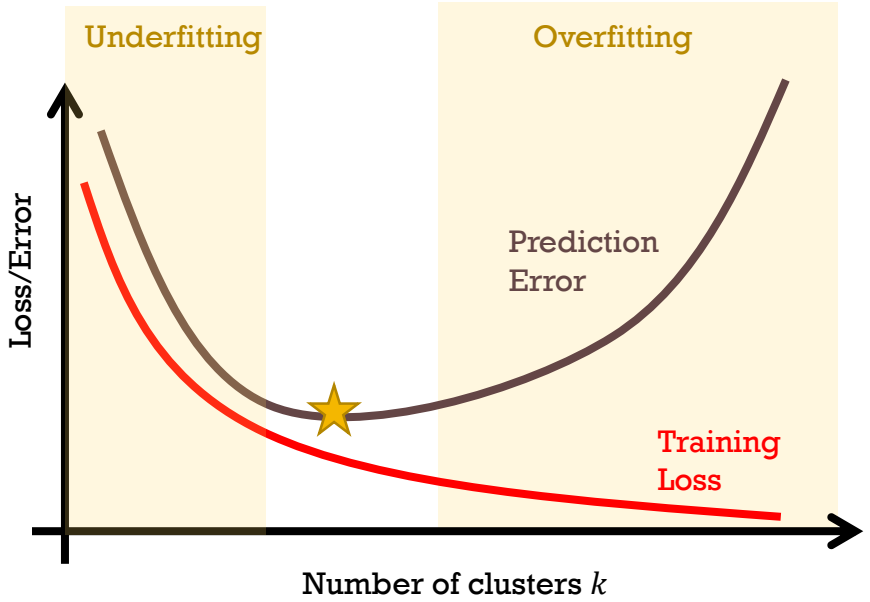


SEMI-SUPERVISED LEARNING

Generalization

Supervised task with small *labeled* data $\mathcal{S}'$

- For each number of clusters k,
  - Perform k–means on *unlabeled* data.
  - Transform $\mathcal{S}'$ using learned clusters e.g. compute distance to each centroid.
  - Use new features for supervised task, and compute prediction error.



- Pick k with smallest prediction error.

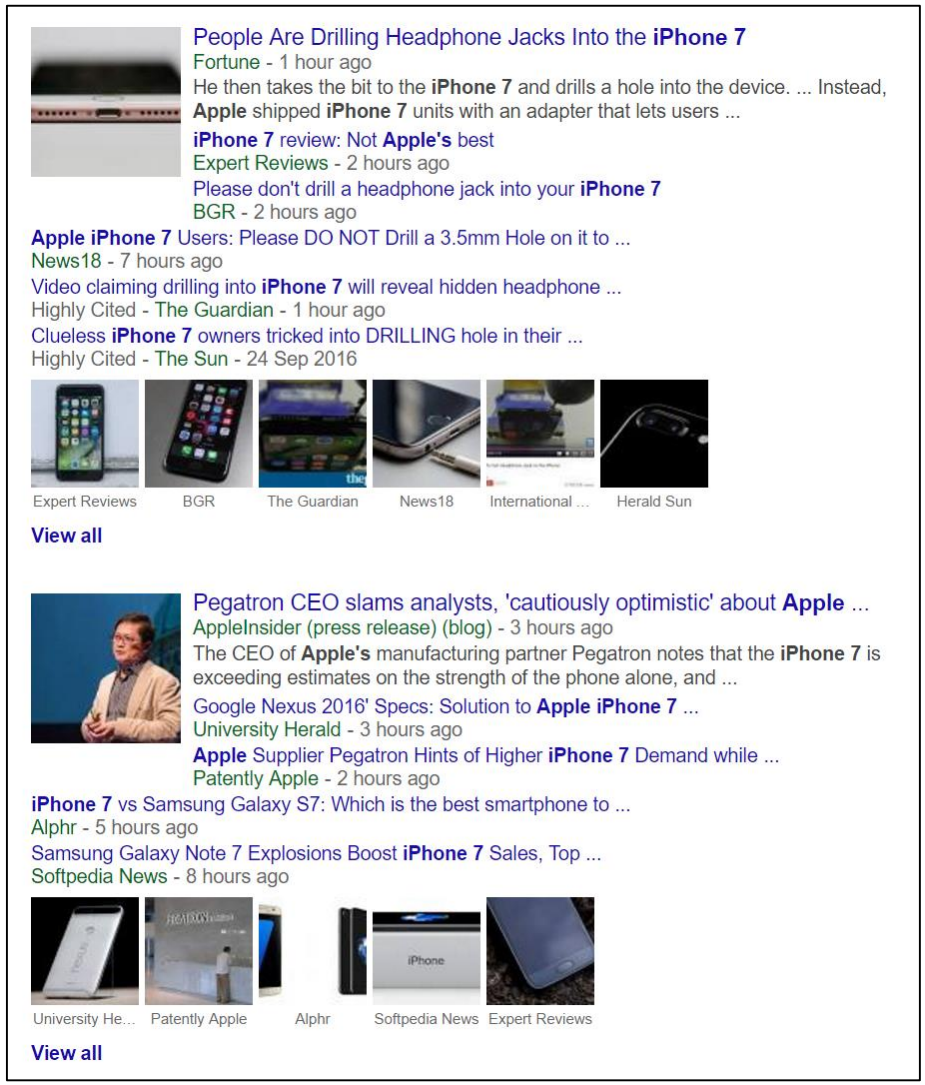
K-MEDROIDS

Use exemplars
instead of centroids.

e.g. Google News.

Repeat until convergence:

- Find best clusters given exemplars
- Find best exemplars given clusters



OPTIMIZATION

* not in syllabus

HOW TO PROGRAM GRADIENT DESCENT

- BFGS (Broyden–Fletcher–Goldfarb–Shanno)
  - Quasi-Newton
  - Homework 2
  https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html
- ADAM (Theano)
  https://gist.github.com/Newmu/acb738767acb4788bac3
- Keras
  https://keras.io/optimizers/
- TensorFlow
  https://www.tensorflow.org/api_docs/python/tf/train/Optimizer

SUMMARY

- Clustering
  - Distance Metric
  - Similarity Functon
  - Training Loss
- Representatives
  - Centroids
  - Exemplars
  - Voronoi Diagrams
- k-Means Algorithm
- Optimization
  - Coordinate Descent
  - Initialization
  - Software
- Generalization
  - Number of Clusters
- Applications
  - Dimensionality Reduction
  - Data Compression
  - Semi-Supervised Learning

INTENDED LEARNING OUTCOMES

Clustering

- Describe the differences between distance metrics and similarity functions. List examples of each of them.
- Write down the training loss using the Euclidean distance.
- Describe two ways of picking representatives for clusters. Explain how Voronoi diagrams are derived from the representatives.
- List two important applications of clustering, and how they are related to dimensionality reduction.

INTENDED LEARNING OUTCOMES

K-Means Algorithm

- Describe the k-means algorithm, and point out how it is based on coordinate descent.
- Explain why it is important to run the k-means algorithm several times at various starting points.
- Describe a procedure for estimating k, the number of clusters.