Unsupervised and Representation Learning

Farzaneh Mirzazadeh

University of California, Santa Cruz

Winter' 17

Most of the slides are from

- Clustering
  - Prof Aarti Singh
    (CMU)
  - Profs Richard Zemel, Raquel Urtasun and Sanja Fidler
    (University of Toronto)
  - Prof David Helmbold
    (UCSC)
- Autoencoders
  Stanford Convolutional Neural Networks for Visual Recognition Course

Unsupervised Learning

Learning from unlabeled, unannotated data, without supervision. Given input X, but not targets y. Some problems to consider:

- how to group data in a meaningful way. Find clusters.
- reduce dimensionality of data
- find hidden causes behind data
- model data density

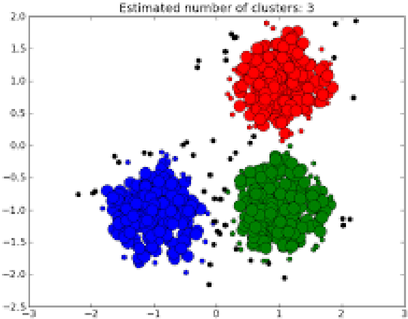
Basic Forms of Unsupervised and Representation Learning

1. Dimensionality reduction
   (unsupervised version of regression, multidimensional target)
2. Clustering
   (unsupervised version of classification, multidimensional target)
3. Coding
   (e.g sparse coding, autoencoders)

Part 1: Clustering

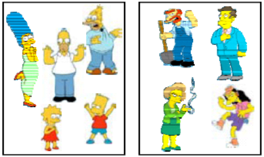
Clustering

- A canonical unsupervised task
- Want to find groups in data such that
  - examples in the same cluster are similar to each other
  - examples in different clusters are not similar to each other

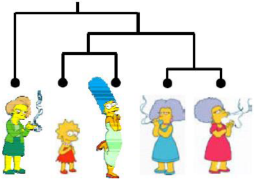


Clustering Algorithms

- Partition algorithms
  - K means clustering
  - Mixture-Model based clustering



- Hierarchical algorithms
  - Single-linkage
  - Average-linkage
  - Complete-linkage
  - Centroid-based



Hierarchical Clustering

- Bottom-Up Agglomerative Clustering

  Starts with each object in a separate cluster, and repeat:

  – Joins the most similar pair of clusters,

  – Update the similarity of the new cluster to other clusters

  until there is only one cluster.

- Top-Down divisive

  Starts with all the data in a single cluster, and repeat:

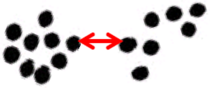
  – Split each cluster into two using a partition based algorithm

  Until each object is a separate cluster.

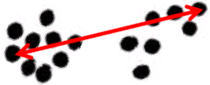
Bottom-up Agglomerative clustering

Different algorithms differ in how the similarities are defined (and hence updated) between two clusters

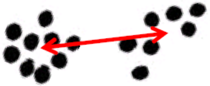
- Single-Link
  - Nearest Neighbor: similarity between their closest members.



- Complete-Link
  - Furthest Neighbor: similarity between their furthest members.



- Centroid
  - Similarity between the centers of gravity



- Average-Link
  - Average similarity of all cross-cluster pairs.

Partitioning Algorithms

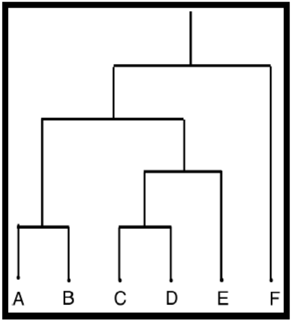
- Partitioning method: Construct a partition of *n* objects into a set of *K* clusters

- Given: a set of objects and the number *K*

- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic method: K-means algorithm

Representing a hierarchical clustering

- ***Dendrogram:*** a tree data structure which illustrates hierarchical clustering techniques.
- Each level shows clusters for that level.
  - Leaf – individual clusters
  - Root – one cluster
- A cluster at level i is the union of its children clusters at level i+1.



Popular clustering algorithm: Kmeans

K-means algorithm

Alternating between centroid selection and assignment stages. Important: Alternating means fix one, learn the other. Fix the other learn one. Until convergence. (expectation maximization)

Clustering time

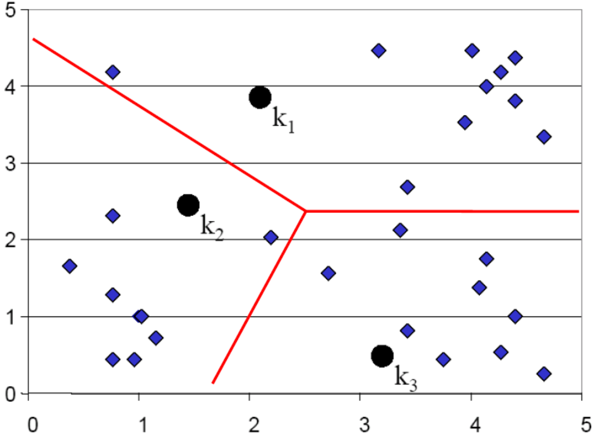
Input: K, t points

1. Initialize K cluster centers randomly
2. While there is a change in cluster assignment
   1. Assign each point to the closest center
   2. Update the cluster center as the mean of already assigned point

Usage time

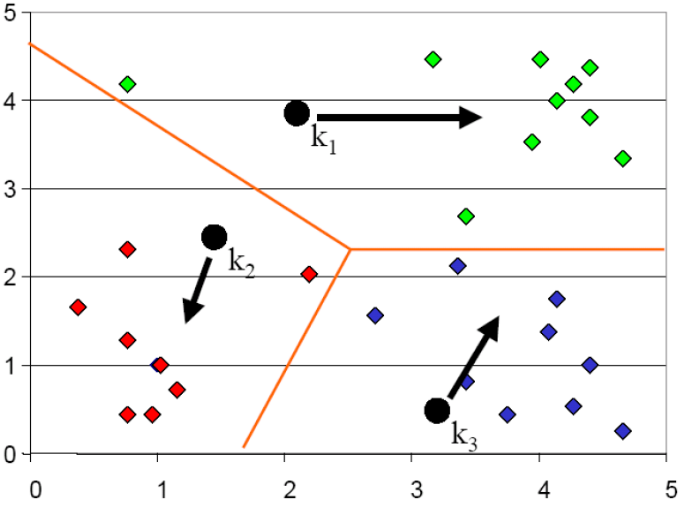
For a new data point $x_{\circ}$ find the cluster with closest center. Assign $x_{\circ}$ to that cluster.

K-means Clustering: Step 1

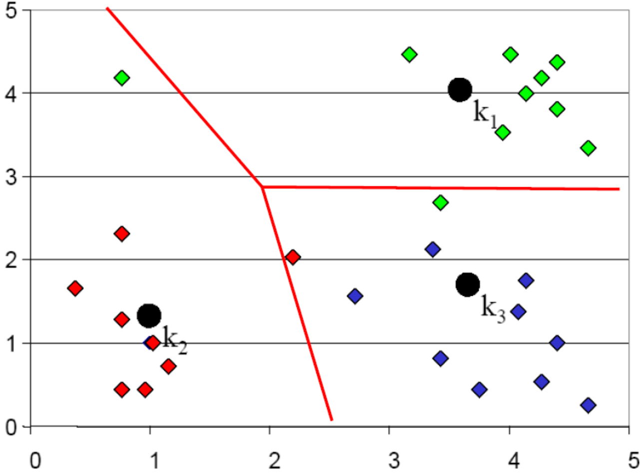


Voronoi diagram

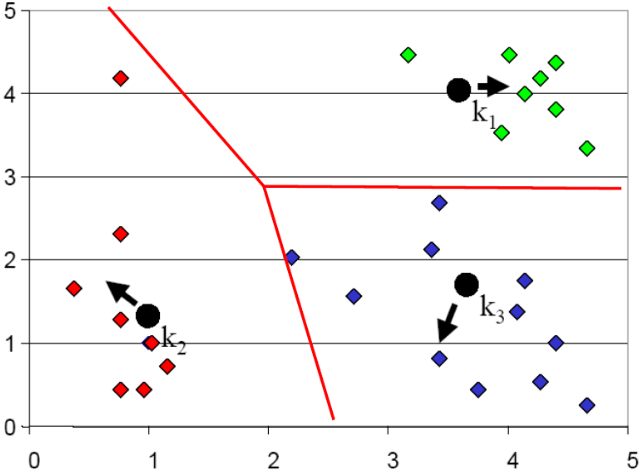
K-means Clustering: Step 2



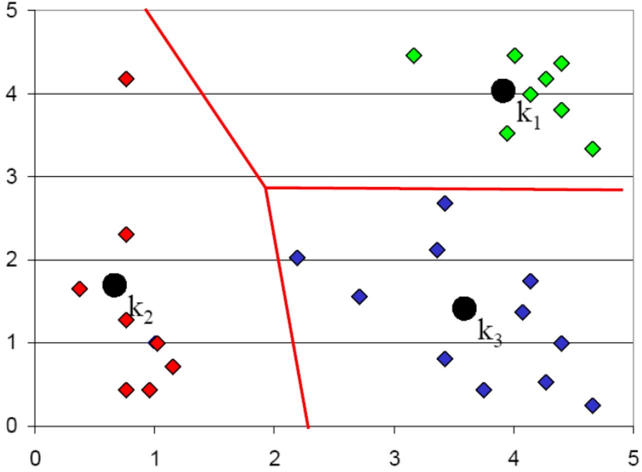
K-means Clustering: Step 3



K-means Clustering: Step 4



K-means Clustering: Step 5



Optimization view to K-means

- Objective function: Distortion Measure

$$\min_{\mu_1,\ldots,\mu_k, r_{ij}} \sum_{j=1}^{k} \sum_{i=1}^{t} r_{ij} \|X_{i:} - \mu_j\|_2^2$$

- Meaning: the goal is to find an assignment of data points to clusters, as well as a set of vectors $\boldsymbol{\mu_j}$ so that the sum of the squares of the distances of each data points to its closes vector $\boldsymbol{\mu_j}$ is minimum.
  - Important: Pay attention to optimization variables. Optimizes wrt both cluster assignment and center of clusters.
  - μ_j is the center of jth cluster.
  - $X_{i:}$ is the ith data point.
  - $r_{ij} \in \{0, 1\}$ indicator that describes which of the K clusters data point $X_{i:}$ is assigned to.
- Finding global min is NP hard.
- How to solve? Find a local min with alternating descent (see next slide)

Alternating descent for solving K-means

- Distortion measure is a function of center locations μ_j and assignments r_{ij}, but not jointly convex in μ_j s and r_{ij} s
- If an objective is jointly convex in a number of variables, it is possible to find global min wrt those variables.
- Here we only can find local min, how? with an iterative procedure called alternating.
- Alternating between two minimization (expectation maximization)
- While not converged, repeat:
  - fix first variable, optimize the objective wrt the second
  - fix the second variable, optimize the objective wrt the first
- Alternating finds a local minimum, if the alternating procedure converges. (i.e. stops)

Alternating for K means

- Phase 1: For fixed cluster centers $\boldsymbol{\mu}_j$, we simply assign the ith data point to the closest cluster center.
- Phase 2: For fixed assignment r, the $\boldsymbol{\mu}_j$ would be equal to the mean of all the data points assigned to cluster j.

Does K-means algorithm stop? Why?

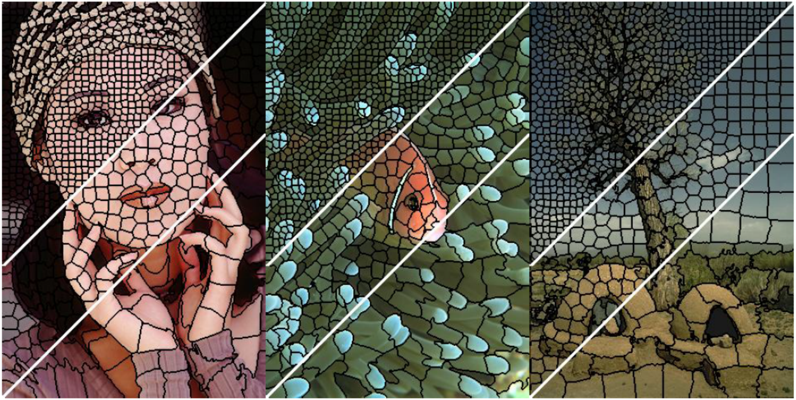
Convergence of K-means algorithm

- Does alternating stop for K-means? YES!
- Why? Based on three observations below:
    1. Distortion is a non-negative functions, so bounded below.
    2. Both steps of alternating in all iterations reduce the distortion measure.
    3. There are only finitely many different assignments of points to clusters (each assignment has a unique center of clusters with it. So finite total combination.

So the procedure has to stop at some point and can not continue reducing the distortion measure forever.

A good candidate question for an exam.

K-means for Image Segmentation



Application example: image segmentation and image compression.
The goal of segmentation is to partition an image into regions with a reasonable homogeneous appearance, or correspond to objects or part of objects.

Numerical example for tracing *K*-means algorithm

- Given: {2,4,10,12,3,20,30,11,25}, k=2
- Randomly assign means: m_1=3,m_2=4
- K_1={2,3}, K_2={4,10,12,20,30,11,25}, m_1=2.5,m_2=16
- K_1={2,3,4},K_2={10,12,20,30,11,25}, m_1=3,m_2=18
- K_1={2,3,4,10},K_2={12,20,30,11,25}, m_1=4.75,m_2=19.6
- K_1={2,3,4,10,11,12},K_2={20,30,25}, m_1=7,m_2=25
- Stop as the clusters with these means stay the same.

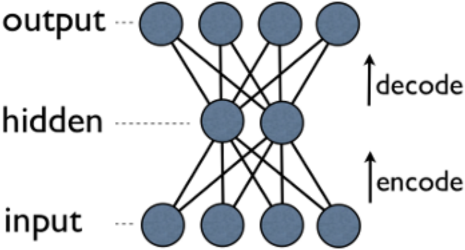
Final notes about K-means

- Different initializations of centers can lead to different clustering. Repeat a couple of times and select the best.
- What number of clusters K is good? A hyper-parameter to set.

Part 2: Auto-encoders
Neural Network for Unsupervised/Representation Learning

Auto-encoder

- A neural network
- The output is its own input.



- Train with back propagation
- Originally one hidden layer
- Can use deep models for encoding and decoding.

Structure of an Autoencoder

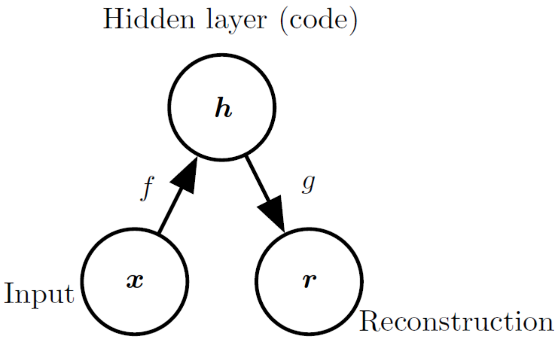
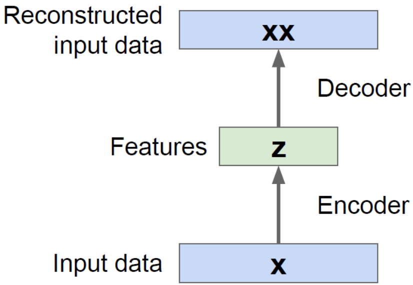
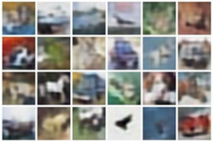


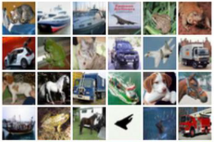
Figure 14.1

(Goodfellow 2016)

Autoencoders



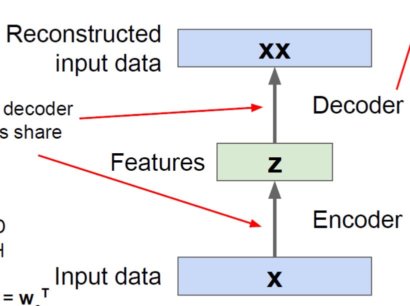




Hidden layer produces: code or features

Autoencoders

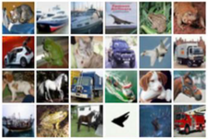
Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN (upconv)



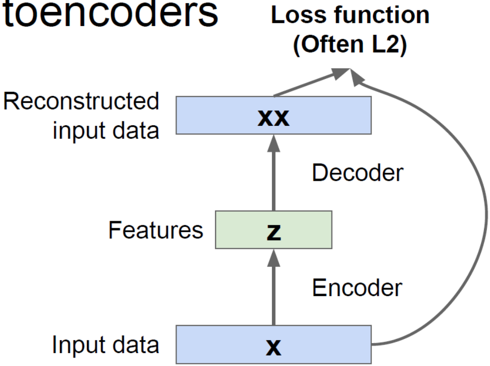
Encoder / decoder sometimes share weights

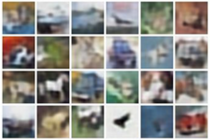
Example:
dim($\mathbf{x}$) = D
dim($\mathbf{z}$) = H
$\mathbf{w}_e$: H x D
$\mathbf{w}_d$: D x H = $\mathbf{w}_e^T$

Train for reconstruction with no labels!

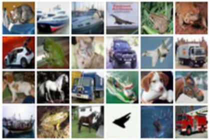


Autoencoders

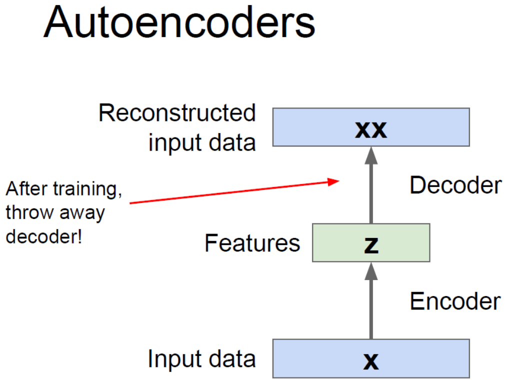


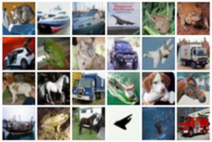


Train for
reconstruction
with no labels!

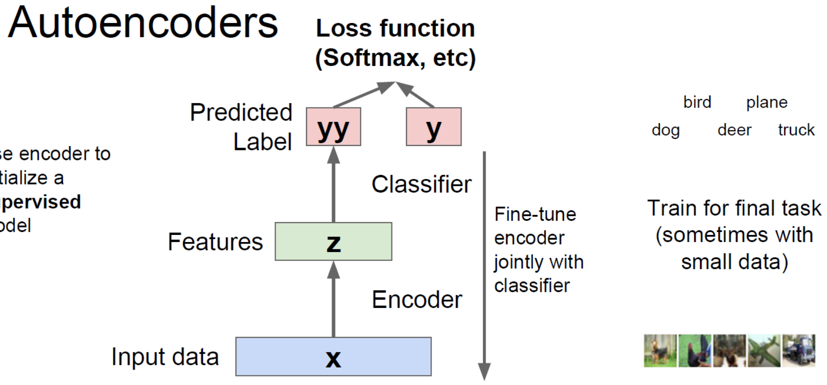


Auto-encoders for pre-training





Auto-encoders for pre-training



Use encoder to initialize a **supervised** model

Summary

Learned

1. Kmeans
2. Auto-encoders