CSC 411: Lecture 12: Clustering

Raquel Urtasun & Rich Zemel

University of Toronto

Oct 28, 2015

Today

- Unsupervised learning
- Clustering
  - k-means
  - Soft k-means

Unsupervised Learning

- Supervised learning algorithms have a clear goal: produce desired outputs for given inputs
- Goal of unsupervised learning algorithms (no explicit feedback whether outputs of system are correct) less clear:
  - Reduce dimensionality
  - Find clusters
  - Model data density
  - Find hidden causes
- Key utility
  - Compress data
  - Detect outliers
  - Facilitate other learning

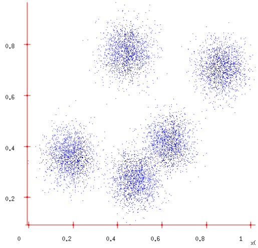
Major types

- Primary problems, approaches in unsupervised learning fall into three classes:

  1. Dimensionality reduction: represent each input case using a small number of variables (e.g., principal components analysis, factor analysis, independent components analysis)
  2. Clustering: represent each input case using a prototype example (e.g., k-means, mixture models)
  3. Density estimation: estimating the probability distribution over the data space

Clustering

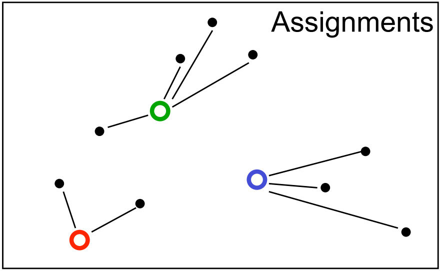
- Grouping N examples into K clusters one of canonical problems in unsupervised learning

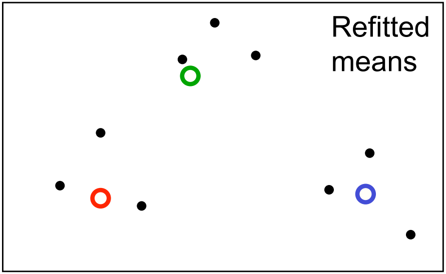


- Motivations: prediction; lossy compression; outlier detection
- We assume that the data was generated from a number of different classes. The aim is to cluster data from the same class together.
  - How many classes?
  - Why not put each datapoint into a separate class?
- What is the objective function that is optimized by sensible clusterings?

The K-means algorithm

- Assume the data lives in a Euclidean space.
- Assume we want k classes/patterns
- Initialization: randomly located cluster centers
- The algorithm alternates between two steps:
  - Assignment step: Assign each datapoint to the closest cluster.
  - Refitting step: Move each cluster center to the center of gravity of the data assigned to it.





K-means Objective

- Objective: minimize sum squared distance of datapoints to their assigned cluster centers

$$\min_{\{\mathbf{m}\},\{\mathbf{r}\}} E(\{\mathbf{m}\},\{\mathbf{r}\}) = \sum_n \sum_k r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

$$\text{s.t.} \sum_k r_k^{(n)} = 1, \forall n, \quad r_k^{(n)} \in \{0,1\}, \forall k, n$$

- Optimization method is a form of coordinate descent ("block coordinate descent")
  - Fix centers, optimize assignments (choose cluster whose mean is closest)
  - Fix assignments, optimize means (average of assigned datapoints)

K-means

- Initialization: Set K means $\{\mathbf{m}_k\}$ to random values
- Assignment: Each datapoint n assigned to nearest mean

$$\hat{k}^n = arg \min_k d(\mathbf{m}_k, \mathbf{x}^{(n)})$$

  and Responsibilities (1 of k encoding)

$$r_k^{(n)} = 1 \longleftrightarrow \hat{k}^{(n)} = k$$

- Update: Model parameters, means, are adjusted to match sample means of datapoints they are responsible for:

$$\mathbf{m}_k = \frac{\sum_n r_k^{(n)} \mathbf{x}^{(n)}}{\sum_n r_k^{(n)}}$$

- Repeat assignment and update steps until assignments do not change

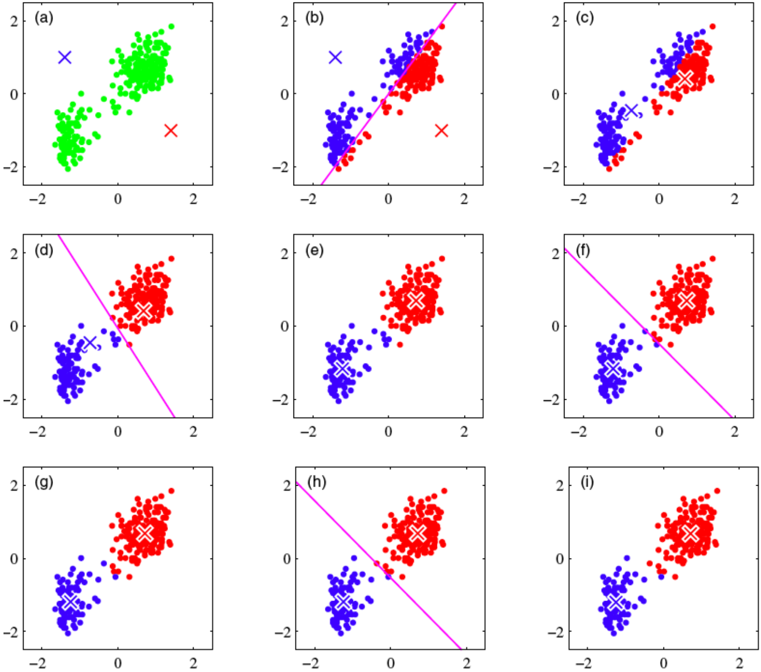


Figure from Bishop

K-means for Image Segmentation and Vector Quantization

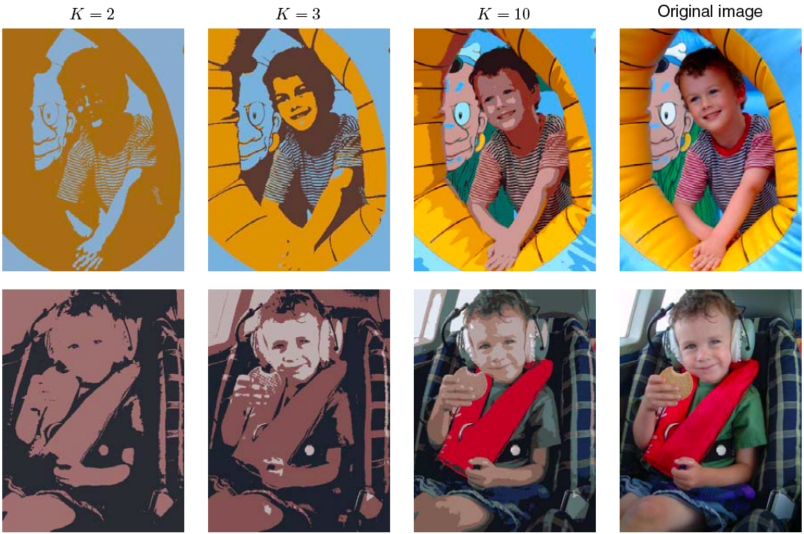
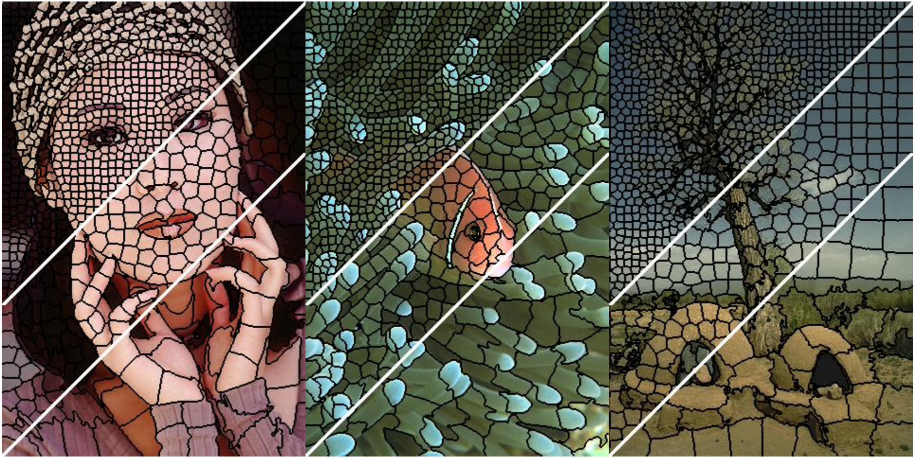


Figure from Bishop

K-means for Image Segmentation



- How would you modify k-means to get super pixels?

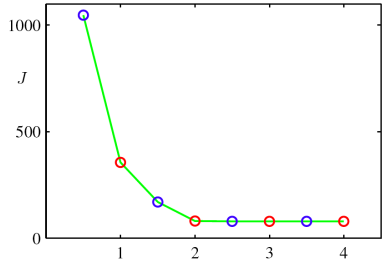
Questions about K-means

- Why does update set $\mathbf{m}_k$ to mean of assigned points?
- Where does distance d come from?
- What if we used a different distance measure?
- How can we choose best distance?
- How to choose K?
- How can we choose between alternative clusterings?
- Will it converge?

Hard cases – unequal spreads, non-circular spreads, inbetween points

Why K-means converges

- Whenever an assignment is changed, the sum squared distances of datapoints from their assigned cluster centers is reduced.
- Whenever a cluster center is moved the sum squared distances of the datapoints from their currently assigned cluster centers is reduced.
- Test for convergence: If the assignments do not change in the assignment step, we have converged (to at least a local minimum).

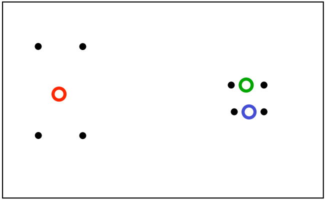


- K-means cost function after each E step (blue) and M step (red). The algorithm has converged after the third M step

Local Minima

- There is nothing to prevent k-means getting stuck at local minima.
- We could try many random starting points
- We could try non-local split-and-merge moves:
  - Simultaneously merge two nearby clusters
  - and split a big cluster into two

A bad local optimum



Soft k-means

- Instead of making hard assignments of datapoints to clusters, we can make soft assignments. One cluster may have a responsibility of .7 for a datapoint and another may have a responsibility of .3.
  - Allows a cluster to use more information about the data in the refitting step.
  - What happens to our convergence guarantee?
  - How do we decide on the soft assignments?

Soft K-means Algorithm

- Initialization: Set K means $\{\mathbf{m}_k\}$ to random values
- Assignment: Each datapoint n given soft "degree of assignment" to each cluster mean k, based on responsibilities

$$r_k^{(n)} = \frac{\exp[-\beta d(\mathbf{m}_k, \mathbf{x}^{(n)})]}{\sum_j \exp[-\beta d(\mathbf{m}_j, \mathbf{x}^{(n)})]}$$

- Update: Model parameters, means, are adjusted to match sample means of datapoints they are responsible for:

$$\mathbf{m}_k = \frac{\sum_n r_k^{(n)} \mathbf{x}^{(n)}}{\sum_n r_k^{(n)}}$$

- Repeat assignment and update steps until assignments do not change

Questions about soft K-means

- How to set β?
- What about problems with elongated clusters?
- Clusters with unequal weight and width

A generative view of clustering

- We need a sensible measure of what it means to cluster the data well.
  - This makes it possible to judge different models.
  - It may make it possible to decide on the number of clusters.
- An obvious approach is to imagine that the data was produced by a generative model.
  - Then we can adjust the parameters of the model to maximize the probability that it would produce exactly the data we observed.