CS 340 Lec. 19: Unsupervised Learning - K-Means

AD

March 2011

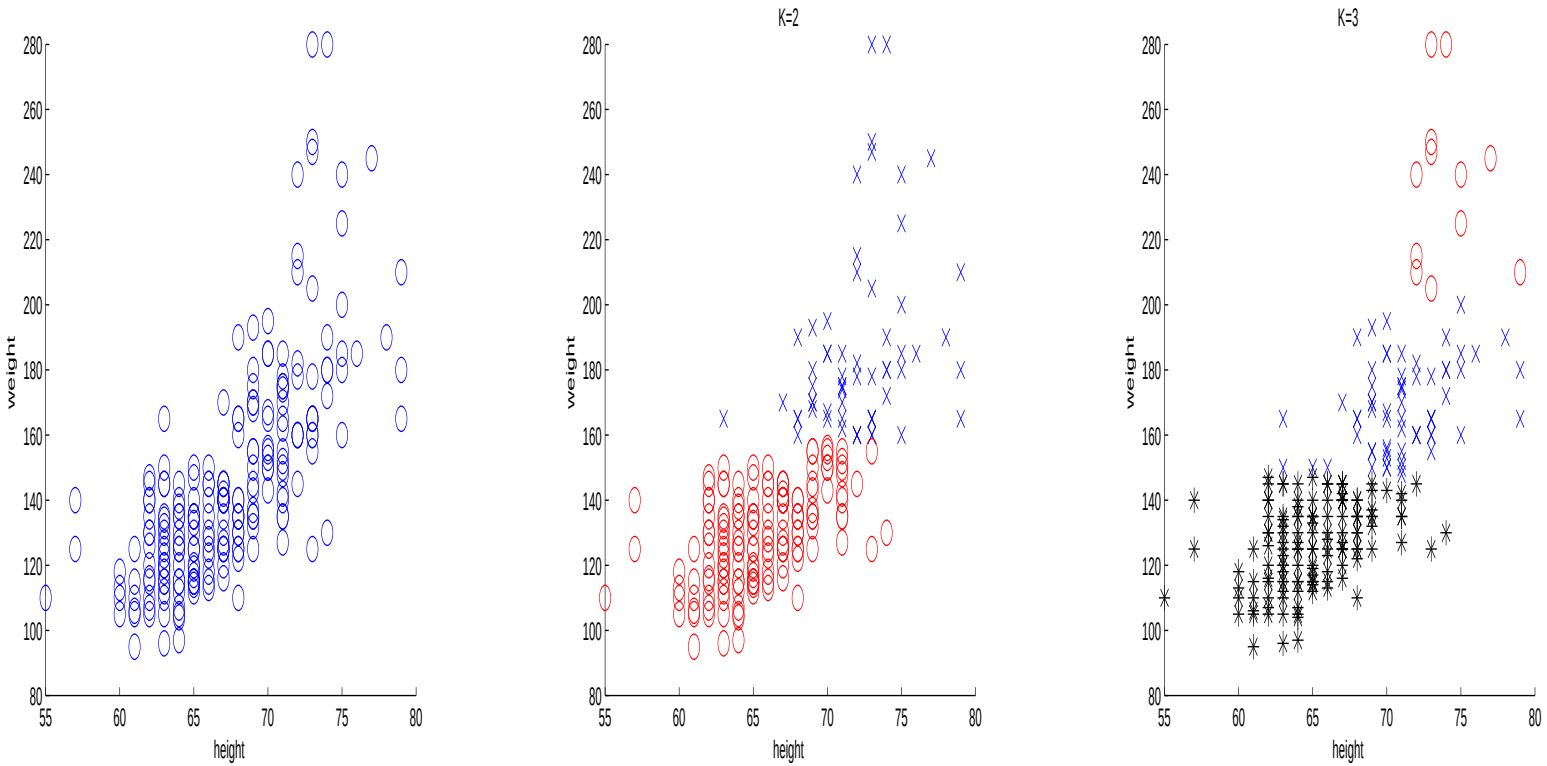
Unsupervised Learning

- In supervised learning, we have training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ and we want to learn how to predict y given a new $\mathbf{x}$.
- In unsupervised learning, we just have data $\{\mathbf{x}_i\}_{i=1}^N$.
- Our goal is to "summarize" or find "patterns" or "structure" in the data using clustering, density estimation and dimensionality reduction.
- The definition of "ground truth" is often missing: no clear error function, or at least many reasonable alternatives
- Useful in exploratory data analysis, and as a pre-processing step for supervised learning

Clustering

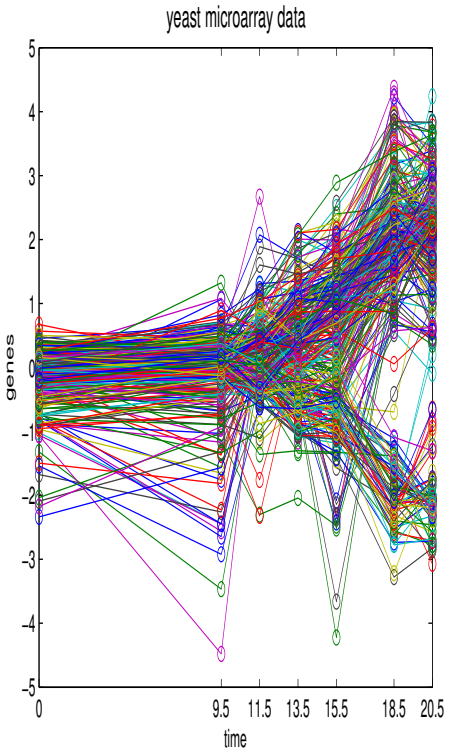
- Clustering is grouping similar objects together.
  - To simplify data for further analysis/learning.
  - To establish prototypes, or detect outliers.
  - To visualize data (in conjunction with dimensionality reduction).
- Clusterings are usually not "right" or "wrong" – different clusterings/clustering criteria can reveal different things about the data.
- Clustering algorithms:
  - Employ some notion of distance/measure of similarity between objects.
  - Have an explicit or implicit criterion defining what a good cluster is and optimize this criterion to determine the clustering.

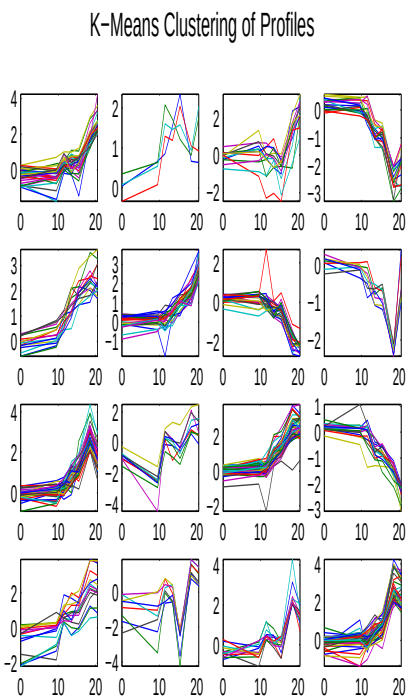
Clustering Height and Weight of Some People

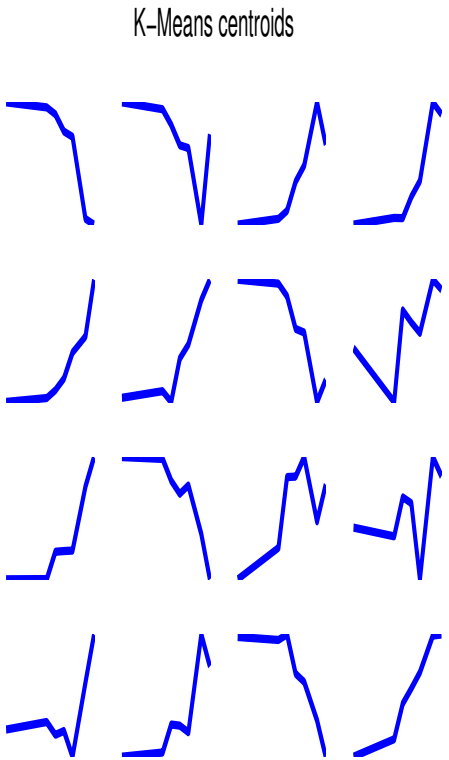


(left) height and weight of some people (center) a possible clustering with $K = 2$ clusters (right) a possible clustering using $K = 3$ clusters

Clustering Yeast Gene Expression Data







(left) Yeast gene expression data plotted as time series (center) a possible clustering with $K = 16$ clusters (right) Cluster centers.

K-Means Clustering

- One of the most popular clustering algorithms: easy to implement and fast.
- Assume the data $\{\mathbf{x}_i\}_{i=1}^N$ to be clustered are d-dimensional real-vectors.
- **Goal**: Find
  - *cluster labels*: $\{z_i\}_{i=1}^N$ where each $z_i \in \{1, 2, ..., K\}$.
  - *cluster centers*: $\{\boldsymbol{\mu}_k\}_{k=1}^K$ where each $\boldsymbol{\mu}_k \in \mathbb{R}^d$.
- We will always have

$$z_i = \underset{k \in \{1,...,K\}}{\arg\min} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

  each point is assigned to the closest cluster centers.

K-Means Clustering Algorithm

- **Initialization**, $t = 0$. Set $\left\{ z_i^{(0)} \right\}_{i=1}^{N}$ to some initial values (e.g. random initial values)
- **At iteration** t, $t \geq 1$.
  - Update the cluster centers. For $k = 1, ..., K$ set

$$\boldsymbol{\mu}_k^{(t)} = \frac{\sum_{i=1}^{N} \mathbf{x}_i \, \mathbb{I}\left(z_i^{(t-1)} = k \right)}{\sum_{i=1}^{N} \mathbb{I}\left(z_i^{(t-1)} = k \right)};$$

    i.e. $\boldsymbol{\mu}_k^{(t)}$ is the mean of all the data assigned to the cluster.
  - For $i = 1, ..., N$, set $z_i^{(t)} = \underset{k \in \{1,...,K\}}{\arg\min} \left\| \mathbf{x}_i - \boldsymbol{\mu}_k^{(t)} \right\|^2$.
- The algorithm converges in finite time and provide an estimate of cluster centers $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ and cluster labels $\{z_i\}_{i=1}^{N}$.

K-Means Clustering Objective Function

- The K-Means clustering algorithm seeks to minimize

$$J\left(\{z_i\}_{i=1}^{N}, \{\boldsymbol{\mu}_k\}_{k=1}^{K}\right) = \sum_{i=1}^{N} \left\|\mathbf{x}_i - \boldsymbol{\mu}_{z_i}\right\|^2 .$$

- This objective function can take K^N possible values and K-means is a greedy algorithm which finds a *local minimum* of J.
- Each time we reassign a vector to a cluster with a nearer centroid, J decreases (or stays the same.).
- Each time we recompute the centroids of each cluster, J decreases (or stays the same.)
- Thus, the algorithm must terminate but the solution depends on the initial assignments of clusters. Different initializations might give different solutions.

Initialization Recipe...

- Assigning each item to random cluster in $\{1, ..., K\}$ is sensible but typically results in cluster centroids near the centroid of all the data in the first round.
- A different heuristic tries to spread the initial centroids around as much as possible:
  - Place first center on top of a randomly chosen data point
  - Place second center on a data point as far away as possible from the first one
  - Place the i-th center as far away as possible from the closest of centers 1 through $i - 1$
- K-means clustering typically runs quickly. With a randomized intialization step, you can run the algorithm multiple times and take the clustering with smallest J.

Example application: Color quantization

- Suppose you have an image stored with 24 bits per ($\approx$ 17 millions colors) pixel and want to compress it so that you use only K colors.
- You want the compressed image to look as similar as possible to the original image
- Perform K-means clustering on the original set of color vectors with K colors.
- Cluster centers (rounded to integer intensities) form the entries in the K-color colormap.

Example application: Color quantization

More generally: Vector quantization with Euclidean loss

- Suppose we want to send all the instances over a communication channel
- In order to compress the message, we cluster the data and encode each instance as the center of the cluster to which it belongs
- The reconstruction error for real-valued data can be measured as Euclidian distance between the true value and its encoding.
- An optimal K-means clustering minimizes the reconstruction error among all possible codings of the same type

How to Select K?

- In quantization/compression applications, K is fixed but in most other applications we would like to determine it from the data.
- Without a probabilistic model, it is difficult to have a sensible procedure here: cross-validation is not applicable here!
- Heuristic Ideas:
  - Delete clusters that cover too few points.
  - Split clusters that cover too many points.
  - Add extra clusters for "outliers".