k-Means Clustering

He He
Slides based on Lecture 13a from David Rosenberg's course materials
(https://github.com/davidrosenberg/mlcourse)

CDS, NYU

April 27, 2021

Unsupervised learning

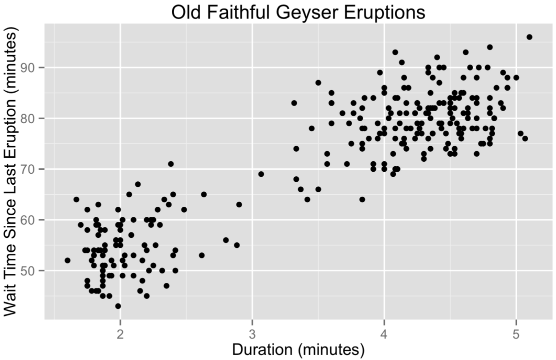
Goal Discover interesting *structure* in the data.

Formulation Density estimation: $p(x;\theta)$ (often with *latent* variables).

Examples

- Discover *clusters*: cluster data into groups.
- Discover *factors*: project high-dimensional data to a small number of "meaningful" dimensions, i.e. dimensionality reduction.
- Discover *graph structures*: learn joint distribution of correlated variables, i.e. graphical models.

Example: Old Faithful Geyser



- Looks like two clusters.
- How to find these clusters algorithmically?

k-Means: By Example

- Standardize the data.
- Choose two cluster centers.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(a).

k-means: by example

- Assign each point to closest center.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(b).

k-means: by example

- Compute new cluster centers.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(c).

k-means: by example

- Assign points to closest center.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(d).

k-means: by example

- Compute cluster centers.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(e).

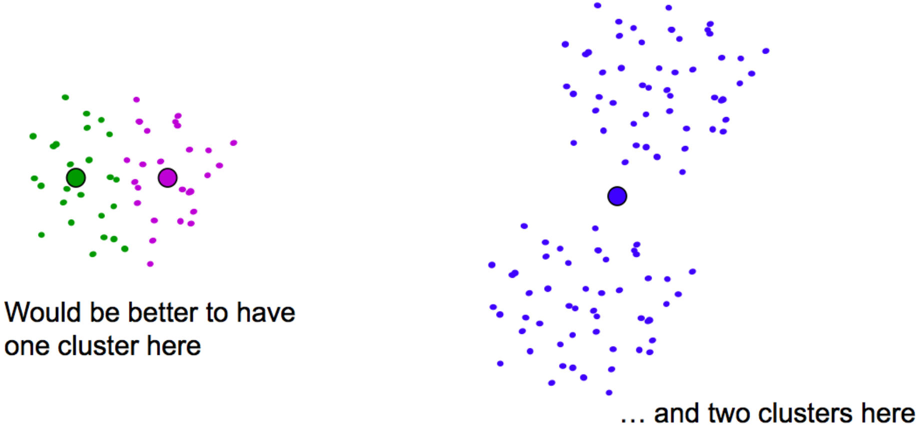
k-means: by example

- Iterate until convergence.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(i).

Suboptimal Local Minimum

- The clustering for $k = 3$ below is a local minimum, but suboptimal:



From Sontag's DS-GA 1003, 2014, Lecture 8.

Formalize k-Means

- Dataset $\mathcal{D} = \{x_1, \ldots, x_n\} \subset \mathcal{X}$ where $\mathcal{X} = \mathbf{R}^d$.
- Goal: Partition data $\mathcal{D}$ into k disjoint sets $C_1, \ldots, C_k$.
- Let $c_i \in \{1, \ldots, k\}$ be the cluster assignment of x_i.
- The **centroid** of C_i is defined to be

$$\mu_i = \underset{\mu \in \mathcal{X}}{\arg\min} \sum_{x \in C_i} \|x - \mu\|^2. \qquad \text{mean of } C_i \tag{1}$$

- The k-means objective is to minimize the distance between each example and its cluster centroid:

$$J(c, \mu) = \sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2. \tag{2}$$

k-Means: Algorithm

1. Initialize: Randomly choose initial centroids $\mu_1, \ldots, \mu_k \in \mathbf{R}^d$.
2. Repeat until convergence (i.e. c_i doesn't change anymore):
   1. For all i, set

$$c_i \leftarrow \arg\min_j \|x_i - \mu_j\|^2. \qquad \text{Minimize } J \text{ w.r.t. } c \text{ while fixing } \mu \tag{3}$$

   2. For all j, set

$$\mu_j \leftarrow \frac{1}{|C_j|} \sum_{x \in C_j} x. \qquad \text{Minimze } J \text{ w.r.t. } \mu \text{ while fixing } c. \tag{4}$$

- Recall the objective: $J(c, \mu) = \sum_{i=1}^n \|x_i - \mu_{c_i}\|^2$.

Avoid bad local minima

k-means converges to a local minimum.

- J is non-convex, thus no guarantee to converging to the global minimum.

Avoid getting stuck with bad local minima:

- Re-run with random initial centroids.
- k-**means++**: choose initial centroids that spread over all data points.
  - Randomly choose the first centroid from the data points $\mathcal{D}$.
  - Sequentially choose subsequent centroids from points that are farther away from current centroids:
    - Compute distance between each x_i and the closest already chosen centroids.
    - Randomly choose next centroid with probability proportional to the computed distance squared.

Summary

We've seen

- Clustering—an unsupervised learning problem that aims to discover group assignments.
- k-means:
  - Algorithm: alternating between assigning points to clusters and computing cluster centroids.
  - Objective: minmizing some loss function by cooridinate descent.
  - Converge to a local minimum.

Next, probabilistic model of clustering.

- A generative model of x.
- Maximum likelihood estimation.