Clustering, Mixtures, and the EM Algorithm

Machine Learning: Module 1

Sean Norton; Simon Hoellerbauer

September 19, 2018

The Problem: Finding Groups

In social science, we often believe our observations have some sort of group structure.

- Regime types
- Types of voters
- Types of legislators

However, our data doesn't (generally) come with these groupings conveniently pre-labeled.

E.g. Regime Types

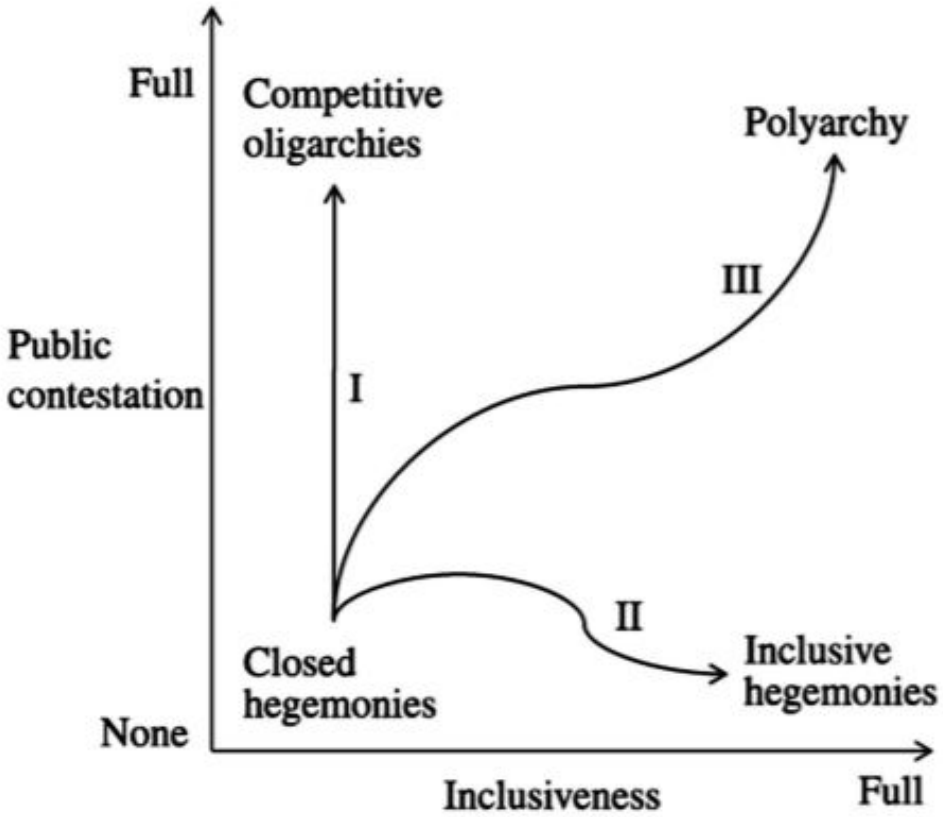


Figure: Dahl: Regime Types

The Problem Continued

The general approach to dealing with this problem has been to hand-label cases. This is problematic because:

- It's time-consuming
- Humans aren't able to consider all available data at once
- It relies on researcher discretion
- For large datasets (e.g. all countries in the world), this is impossible.

What if there was a way to find latent groupings between our cases quickly and with as little researcher discretion as possible?

Enter Cluster Analysis

This is exactly what cluster analysis is intended to do!
Given:

- Data
- Number of clusters
- Variables
- Similarity measure

A cluster analysis algorithm finds groupings, or clusters, that maximize the similarity between observations within a cluster.

K-Means and Notation

One of the most common similarity measures is the squared distance between the center (mean) of a cluster.
This is known as *k-means* or *k-nearest neighbor* clustering.
Before we dive into the math, some notation:

- k: total number of clusters
- r_{nk}: indicator vector of cluster membership for observation x_n
- μ_k: the centroid of cluster k

K-Means: The Math

Cluster analysis relies on a measure of similarity, which in k-means is:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

This number J is also known as a *distortion measure*.
What (hopefully) familiar thing does this measure look like?

The Math cont.

Q: But how do we choose values of μ_k given that we don't actually know the cluster assignments?
A: We don't!
We can find k_m through a version of the *expectation-maximization algorithm*:

1. Initialize some random values for μ_k
2. Minimize J w.r.t. r_{nk}; i.e. assign cluster memberships in order to minimize distortion
3. Using the previous r_{nk}, minimize J w.r.t to μ_k; i.e., assign new means that minimize distortion
4. Repeat until convergence (J does not change, or the change falls below some threshold)

The Math: Visualized

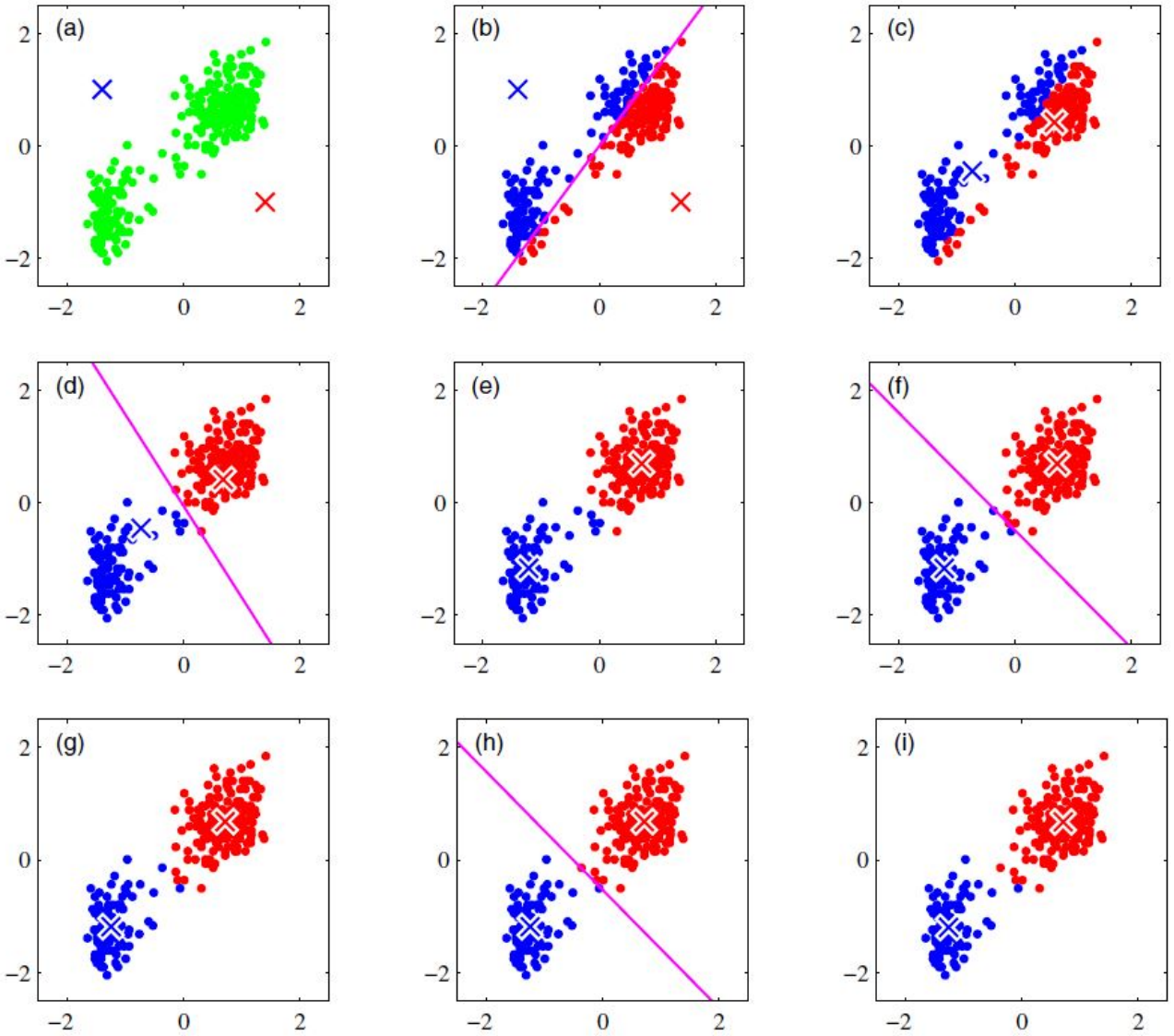


Figure: Optimizing Clusters

Problems with Clustering

- Choosing k: this requires trying a lot of different k w.r.t. some similarity measure
  - Plot the WCSS against the number of clusters, and look for a "bend" in the plot; this is known as the elbow method
  - Use average silhouette width: the silhouette is a measure how similar an observation is to its own cluster (consistency) and how dissimilar it is to other clusters (separation)
  - Gap statistics: compare multiple values of k to a simulated reference distribution of datasets with clusters varying from $k = 1$ to $k = max$
- Sensitivity to outliers; luckily, there are clustering methods other than k-means
  - Partitioning around medoids (PAM): uses median instead of mean
  - Hierarchical clustering: creates a tree-based representation of the data without specifying k; clusters are created by "cutting" the tree.
- Fundamentally descriptive; clusters will not necessarily be the same given different data

Problems cont.

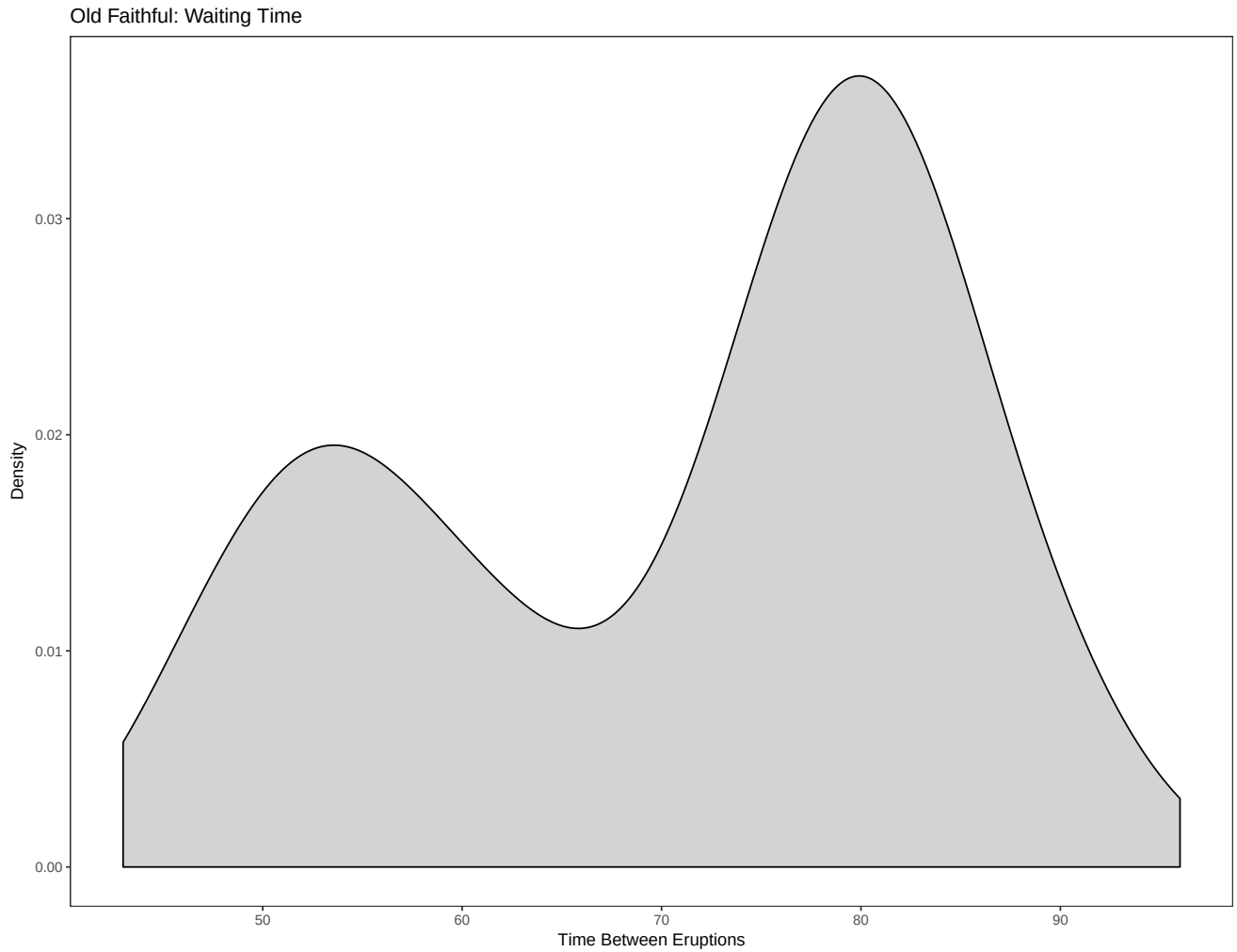
Figure: My Cluster Plot

Mixture Models: Motivation

What if instead of "hard" cluster assignments, we could assign a probability that cases belong to a particular group?
That is precisely what *mixture models* are intended to do!
Mixture models allow observations to belong to different distributions - either different parameterizations of the same distribution, or different distributions entirely.
This allows us to:

- Sort cases into groups, much like cluster models, but do so in a way that quantifies uncertainty
- Estimate a model in the same step as the grouping, which most out-of-the-box clustering packages do not do
- Work with data that has a multimodal distribution without having to discard a substantial amount of variation.

Motivating Example: Old Faithful



Motivating Example: Old Faithful

Clearly, this data appears to be normally distributed, but not in a single normal distribution.
Instead of fitting a single Gaussian, we can fit a *mixture* of Gaussians of the following form:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

Where π_k is the probability that an observation belongs to cluster k.

Deriving the Model

To be able to estimate π_k, let us define a K-dimensional vector $\mathbf{z}$, which is 1 if an observation is in group k and 0 otherwise.

$$p(z_k = 1) = \pi_k$$

Or alternatively, since this is a 1-of-K vector:

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$$