Clustering and The Expectation-Maximization Algorithm
Unsupervised Learning

Marek Petrik

3/7

Some of the figures in this presentation are taken from “An Introduction to Statistical Learning, with applications in R” (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
Learning Methods

1. Supervised Learning: Learning a function $f$:

$$Y = f(X) + \epsilon$$

1.1 Regression
1.2 Classification

2. Unsupervised learning: Discover interesting properties of data (no labels)

$$X_1, X_2, \ldots$$

2.1 Dimensionality reduction or embedding
2.2 Clustering
Principal Components Analysis

- Reduce dimensionality
- Start with features $X_1 \ldots X_n$
- Construct fewer features $Z_1 \ldots Z_M$

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p$$

- Weights are usually normalized (using $\ell_2$ norm)

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

- Data has greatest variance along $Z_1$
1st Principal Component

1st Principal Component: Direction with the largest variance

\[ Z_1 = 0.839 \times (\text{pop} - \overline{\text{pop}}) + 0.544 \times (\text{ad} - \overline{\text{ad}}) \]
More Unsupervised Learning: Discovering Structure of Data

1. K-Means Clustering
2. Hierarchical Clustering
3. Expectation-Maximization Method (Not Covered in ISL, see ESL 8.5)
Clustering

Simplify data in a different way than PCA.

- PCA finds a low-dimensional representation of data
- Clustering finds homogeneous subgroups among the observations
Clustering: Assumptions and Goals

- Exists a method for measuring similarity between data points
- Some points are more similar than others

- Discover **latent** patterns that exist but may not be observed/observable
Clustering: Assumptions and Goals

- Exists a method for measuring similarity between data points
- Some points are more similar than others

- Want to identify similarity patterns
  1. Discover the different types of disease
  2. Market segmentation: Types of users that visit a website
  3. Discover movie or book genres
  4. Discover types of topics in documents

- Discover latent patterns that exist but may not be observed/observable
Clustering Algorithms

- **K-Means**: simple and effective
- **Hierarchical clustering**: Many complex clusters
- Many other clustering methods, most heuristics
- **EM**: General algorithm for dealing with latent variables by *maximizing likelihood*
K-Means Clustering

- Cluster data into *complete* and *non-overlapping* sets
- Example:
K-Means Objective

- $k$-th cluster: $C_k$
- $i$-th observation in cluster $k$: $i \in C_k$
K-Means Objective

- $k$-th cluster: $C_k$
- $i$-th observation in cluster $k$: $i \in C_k$
- Find clusters that are homogeneous: $W(C_k)$ homogeneity of clusters

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} W(C_k)$$
K-Means Objective

- $k$-th cluster: $C_k$
- $i$-th observation in cluster $k$: $i \in C_k$
- Find clusters that are homogeneous: $W(C_k)$ homogeneity of clusters

\[
\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} W(C_k)
\]

- Define homogeneity as in-cluster variance

\[
\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \left( \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right)
\]
K-Means Objective

- \( k \)-th cluster: \( C_k \)
- \( i \)-th observation in cluster \( k \): \( i \in C_k \)
- Find clusters that are homogeneous: \( W(C_k) \) homogeneity of clusters

\[
\min_{C_1, \ldots, C_K} \sum_{k=1}^{K} W(C_k)
\]

- Define homogeneity as in-cluster variance

\[
\min_{C_1, \ldots, C_K} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2
\]

- This is an NP hard problem
K-Means Algorithm
Heuristic solution to the minimization problem

1. Randomly assign cluster numbers to observations
2. Iterate while clusters change
   2.1 For each cluster, compute the centroid
   2.2 Assign each observation to the closest cluster

Note that:

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$
K-Means Illustration

Data

Step 1

Iteration 1, Step 2a

Iteration 1, Step 2b

Iteration 2, Step 2a

Final Results
Properties of K-Means

- *Local minimum*: Does not necessarily find the optimal solution
- Multiple runs can result in different solutions
- Choose the result of the run with **minimal objective**
- Cluster labels do not matter
Multiple Runs of K-Means
Hierarchical Clustering

- Multiple levels of similarity needed in complex domains
- Build a similarity tree
Dendrogram: Similarity Tree
Hierarchical Clustering Algorithm

1. Begin with $n$ observations and compute $\binom{n}{2}$ dissimilarity measures
2. For $i = n, n - 1, \ldots, 2$
   2.1 Fuse 2 most similar clusters
   2.2 Update $i - 1$ dissimilarities
Hierarchical Clustering Algorithm: Illustration

![Hierarchical Clustering Diagram]

The diagram illustrates the hierarchical clustering process for a set of data points in two dimensions, $X_1$ and $X_2$. The process involves iterative merging of the closest clusters based on a distance metric, resulting in a dendrogram that can be cut at a certain level to obtain a specific number of clusters.

The first step shows the initial separation of points into individual clusters. In subsequent steps, clusters are merged based on the shortest distance between their centroids or on a predefined linkage criterion (e.g., single linkage, complete linkage, average linkage).

By the final step, all points are grouped into a single cluster, reflecting the complete hierarchy of the clustering process.
Dissimilarity Measure: Linkage

1. Complete
2. Single
3. Average
4. Centroid
Impact of Dissimilarity Measure

Average Linkage

Complete Linkage

Single Linkage
Clustering in Practice

- Fraught with problems: no clear measure of quality (like MSE)
- How to choose $k$? Problem dependent
- Standardize features, center them?
- What dissimilarity to use?
Clustering in Practice

- Fraught with problems: no clear measure of quality (like MSE)
- How to choose $k$? Problem dependent
- Standardize features, center them?
- What dissimilarity to use?
- Careful over-explaining clustering results: source: http://miriamposner.com
Expectation-Maximization

- Maximum likelihood approach to clustering
- General method for dealing with latent features / labels
- Especially useful with generative models
- A heuristic method used to solve complex optimization problems
- Generalization of the idea: Minorization-Maximization
Recall LDA
LDA: Linear Discriminant Analysis

- **Generative model**: capture probability of predictors for each label

- **Predict**:
  1. $\text{Pr(\text{balance} | \text{default} = \text{yes})}$ and $\text{Pr(\text{default} = \text{yes})}$
  2. $\text{Pr(\text{balance} | \text{default} = \text{no})}$ and $\text{Pr(\text{default} = \text{no})}$

Classes are normal: $\text{Pr(\text{balance} | \text{default} = \text{yes})}$
LDA: Linear Discriminant Analysis

- **Generative model**: capture probability of predictors for each label

- **Predict**:
  1. $\Pr[\text{balance} \mid \text{default} = \text{yes}]$ and $\Pr[\text{default} = \text{yes}]$
LDA: Linear Discriminant Analysis

- **Generative model**: capture probability of predictors for each label

- Predict:
  1. $\Pr[\text{balance} \mid \text{default} = \text{yes}]$ and $\Pr[\text{default} = \text{yes}]$
  2. $\Pr[\text{balance} \mid \text{default} = \text{no}]$ and $\Pr[\text{default} = \text{no}]$
LDA: Linear Discriminant Analysis

- **Generative model**: capture probability of predictors for each label

1. $\Pr[\text{balance} \mid \text{default} = \text{yes}]$ and $\Pr[\text{default} = \text{yes}]$
2. $\Pr[\text{balance} \mid \text{default} = \text{no}]$ and $\Pr[\text{default} = \text{no}]$

- Classes are normal: $\Pr[\text{balance} \mid \text{default} = \text{yes}]$
LDA vs Logistic Regression

- **Logistic regressions:**
  \[ \text{Pr}[\text{default} = \text{yes} | \text{balance}] \]

- **Linear discriminant analysis:**
  \[ \text{Pr}[\text{balance} | \text{default} = \text{yes}] \quad \text{and} \quad \text{Pr}[\text{default} = \text{yes}] \]
  \[ \text{Pr}[\text{balance} | \text{default} = \text{no}] \quad \text{and} \quad \text{Pr}[\text{default} = \text{no}] \]
LDA with 1 Feature

- Classes are normal and class probabilities $\pi_k$ are scalars

$$f_k(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( - \frac{1}{2\sigma^2} (x - \mu_k)^2 \right)$$

- **Key Assumption:** Class variances $\sigma_k^2$ are the same.
EM For LDA

- Labels are missing, guess them
- Find the most likely model and latent observations:

\[
\max_{\text{model}} \log \ell(\text{model}) = \max_{\text{model}} \log \sum_{\text{latent}} \Pr[\text{data}, \text{latent} | \text{model}] =
\]
EM For LDA

- Labels are missing, guess them
- Find the most likely model and latent observations:

\[
\max_{\text{model}} \log \ell(\text{model}) = \max_{\text{model}} \log \sum_{\text{latent}} \Pr[\text{data, latent} \mid \text{model}] = \\
= \max_{\text{model}} \log \sum_{\text{latent}} \Pr[\text{data} \mid \text{latent, model}] \Pr[\text{latent} \mid \text{model}]
\]
EM For LDA

- Labels are missing, guess them
- Find the most likely model and latent observations:

\[
\max_{\text{model}} \log \ell(\text{model}) = \max_{\text{model}} \log \sum_{\text{latent}} \Pr[\text{data}, \text{latent} | \text{model}] = \\
= \max_{\text{model}} \log \sum_{\text{latent}} \Pr[\text{data} | \text{latent}, \text{model}] \Pr[\text{latent} | \text{model}]
\]

- Difficult and non-convex optimization problem (log \(\sum\))
EM Derivation

- Iteratively approximate and optimize the log-likelihood function
  1. Construct a concave lower bound
  2. Maximize the lower bound
  3. Repeat

- Notation:
  - Model: $\theta$
  - Data: $x$
  - Latent variables: $z$

\[
\max_{\theta, z} \log \ell(\theta, z) = \max_{\theta, z} \log \Pr[x \mid \theta] = \\
= \max_{\theta, z} \log \sum_z \Pr[x \mid z, \theta] \Pr[z \mid \theta]
\]
EM Derivation

- Suppose we have an estimate of the model $\theta_n$
- How to compute $\theta_{n+1}$ that improves on it?

$$\theta_{n+1}, z_{n+1} = \arg \max_{\theta, z} \log \sum_z \Pr[x, z | \theta] = \arg \max_{\theta, z} \log \sum_z \Pr[z | \theta] \Pr[z | x, \theta] =$$

$$= \arg \max_{\theta, z} \log \sum_z \Pr[z | \theta] \Pr[z | x, \theta] \frac{\Pr[z | x, \theta_n]}{\Pr[z | x, \theta_n]} =$$

$$= \arg \max_{\theta, z} \log \sum_z \Pr[z | x, \theta_n] \frac{\Pr[z | \theta] \Pr[z | x, \theta]}{\Pr[z | x, \theta_n]} \leq$$

$$\leq \arg \max_{\theta, z} \sum_z \Pr[z | x, \theta_n] \log \frac{\Pr[z | \theta] \Pr[z | x, \theta]}{\Pr[z | x, \theta_n]} =$$

$$= \arg \max_{\theta, z} \sum_z \Pr[z | x, \theta_n] \log \Pr[x, z | \theta]$$
EM Algorithm

1. **E Step**: Estimate $\Pr[z \mid x, \theta_n]$ for all values of $z$. (Construct the lower bound)

2. **M-Step**: Maximize the lower bound:

$$\theta_{n+1} = \arg \max_{\theta} \sum_z \Pr[z \mid x, \theta_n] \log \Pr[x, z \mid \theta]$$

This can be solved using traditional MLE methods with weighted samples
EM for Mixture of Gaussians

Rough sketch

1. Randomly assign cluster weights to observations
2. Iterate while clusters change
   2.1 For each cluster, compute the centroid based on observation weights of observations
   2.2 Assign each observation new cluster weights based on the distances from centroids
Other Applications of EM

- Very powerful and general idea!
- Training with missing data for many model types
- Hidden variables in Bayesian nets
- Identifying confounding variables
- Solving difficult (complex) optimization problem: MM