Introduction to Machine Learning

Clustering

Varun Chandola

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
chandola@buffalo.edu

University at Buffalo
Department of Computer Science and Engineering
School of Engeering and Applied Sciences

Outline

Publishing a Magazine

- Imagine your are a magazine editor
- Need to produce the next issue
- What do you do?

Publishing a Magazine

- Imagine your are a magazine editor
- Need to produce the next issue
- What do you do?
  - Call your four assistant editors
    1. Politics
    2. Health
    3. Technology
    4. Sports
  - Ask each to send in k articles
  - Join all to create an issue

Treating a Magazine Issue as a Data Set

- Each article is a data point consisting of words, etc.
- **Each article has a (hidden) type - sports, health, politics, and technology**

Now imagine your are the reader

- Can you assign the type to each article?

Treating a Magazine Issue as a Data Set

- Each article is a data point consisting of words, etc.
- **Each article has a (hidden) type - sports, health, politics, and technology**

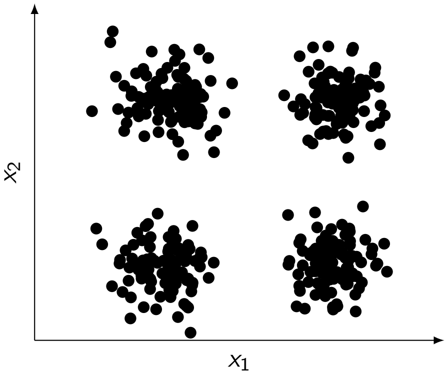
Now imagine your are the reader

- Can you assign the type to each article?
- Simpler problem: **Can you group articles by type?**
- Clustering

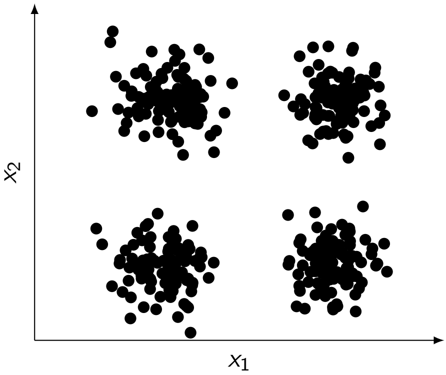
What is Clustering?

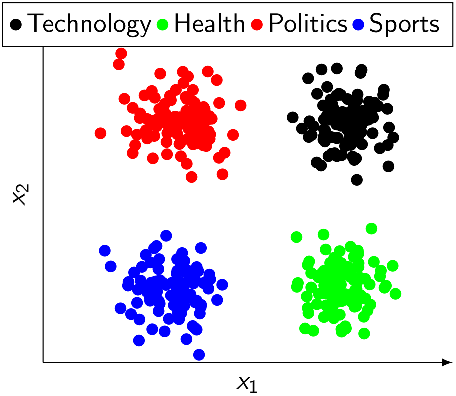
- Grouping similar things together
- A notion of a similarity or distance metric
- A type of **unsupervised learning**
  - Learning without any labels or target

Expected Outcome of Clustering



Expected Outcome of Clustering





K-Means Clustering

- **Objective**: Group a set of N points ($\in \Re^D$) into K clusters.

K-Means Clustering

- **Objective**: Group a set of N points ($\in \Re^D$) into K clusters.

1. **Start** with k *randomly initialized* points in D dimensional space
   - Denoted as $\{\mathbf{c}_k\}_{k=1}^K$
   - Also called *cluster centers*

K-Means Clustering

- **Objective**: Group a set of N points ($\in \Re^D$) into K clusters.

1. **Start** with k *randomly initialized* points in D dimensional space
   - Denoted as $\{\mathbf{c}_k\}_{k=1}^K$
   - Also called *cluster centers*
2. **Assign** each input point $\mathbf{x}_n$ ($\forall n \in [1, N]$) to cluster k, such that:

$$\min_k \text{dist}(\mathbf{x}_n, \mathbf{c}_k)$$

K-Means Clustering

- **Objective**: Group a set of N points ($\in \Re^D$) into K clusters.

1. **Start** with k *randomly initialized* points in D dimensional space
   - Denoted as $\{\mathbf{c}_k\}_{k=1}^K$
   - Also called *cluster centers*
2. **Assign** each input point $\mathbf{x}_n$ ($\forall n \in [1, N]$) to cluster k, such that:

$$\min_k \text{dist}(\mathbf{x}_n, \mathbf{c}_k)$$

3. **Revise** each cluster center $\mathbf{c}_k$ using all points assigned to cluster k

K-Means Clustering

- **Objective**: Group a set of N points ($\in \Re^D$) into K clusters.

1. **Start** with k *randomly initialized* points in D dimensional space
   - Denoted as $\{\mathbf{c}_k\}_{k=1}^{K}$
   - Also called *cluster centers*
2. **Assign** each input point $\mathbf{x}_n$ $(\forall n \in [1, N])$ to cluster k, such that:

$$\min_k \text{dist}(\mathbf{x}_n, \mathbf{c}_k)$$

3. **Revise** each cluster center $\mathbf{c}_k$ using all points assigned to cluster k
4. **Repeat** 2

Variants of K-Means

- Finding distance
  - Euclidean distance is popular
- Finding cluster centers
  - Mean for K-Means
  - Median for k-medoids

Choosing Parameters

1. Similarity/distance metric
   - Can use non-linear transformations
   - K-Means with Euclidean distance produces "circular" clusters
2. How to set k?
   - Trial and error
   - How to evaluate clustering?
   - K-Means objective function

$$J(\mathbf{c}, \mathbf{R}) = \sum_{n=1}^{N} \sum_{k=1}^{K} R_{nk} \|\mathbf{x}_n - \mathbf{c}_k\|^2$$

   - $\mathbf{R}$ is the cluster assignment matrix

$$R_{nk} = \begin{cases} 1 & \text{If } \mathbf{x}_n \in \text{ cluster } k \\ 0 & \text{Otherwise} \end{cases}$$

Initialization Issues

- Can lead to wrong clustering
- Better strategies
  1. Choose first centroid randomly, choose second farthest away from first, third farthest away from first and second, and so on.
  2. Make multiple runs and choose the best

Strengths and Limitations of K-Means

Strengths

- Simple
- Can be extended to other types of data
- Easy to parallelize

Weaknesses

- Circular clusters (not with kernelized versions)
- Choosing K is always an issue
- Not guaranteed to be optimal
- Works well if natural clusters are round and of equal densities
- **Hard Clustering**

Issues with K-Means

- "Hard clustering"
- Assign every data point to exactly one cluster
- **Probabilistic Clustering**
  - Each data point can belong to multiple clusters with varying probabilities
  - In general

$$P(\mathbf{x}_i \in C_j) > 0 \quad \forall j = 1 \ldots K$$

  - For hard clustering probability will be 1 for one cluster and 0 for all others

Spectral Clustering

- An alternate approach to clustering
- Let the data be a set of N points

$$\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$$

- Let $\mathbf{S}$ be a $N \times N$ **similarity matrix**

$$S_{ij} = sim(\mathbf{x}_i, \mathbf{x}_j)$$

- $sim(,)$ is a similarity function
- Construct a weighted undirected graph from $\mathbf{S}$ with adjacency matrix, $\mathbf{W}$

$$W_{ij} = \begin{cases} sim(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_i \text{ is nearest neighbor of } \mathbf{x}_j \\ 0 & otherwise \end{cases}$$

- Can use more than 1 nearest neighbors to construct the graph

Spectral Clustering as a Graph Min-cut Problem

- Clustering $\mathbf{X}$ into K clusters is equivalent to finding K cuts in the graph $\mathbf{W}$
  - $A_1, A_2, \ldots, A_K$
- Possible objective function

$$cut(A_1, A_2, \ldots, A_K) \triangleq \frac{1}{2}\sum_{k=1}^{K} W(A_k, \bar{A}_k)$$

- where $\bar{A}_k$ denotes the nodes in the graph which are **not in** A_k and

$$W(A, B) \triangleq \sum_{i \in A, j \in B} W_{ij}$$

Straight min-cut results in trivial solution

- For $K = 2$, an optimal solution would have only one node in A_1 and rest in A_2 (or vice-versa)

Normalized Min-cut Problem

$$normcut(A_1, A_2, \ldots, A_K) \triangleq \frac{1}{2}\sum_{k=1}^{K}\frac{W(A_k, \bar{A}_k)}{vol(A_k)}$$

where $vol(A) \triangleq \sum_{i \in A} d_i$, d_i is the weighted degree of the node i

NP Hard Problem

- Equivalent to solving a 0-1 knapsack problem
- Find N binary vectors, $\mathbf{c}_i$ of length K such that $c_{ik} = 1$ only if point i belongs to cluster k
- If we relax constraints to allow c_{ik} to be real-valued, the problem becomes an eigenvector problem
    - Hence the name: **spectral clustering**

The Graph Laplacian

$$\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$$

- $\mathbf{D}$ is a diagonal matrix with degree of corresponding node as the diagonal value

Properties of Laplacian Matrix

1. Each row sums to 0
2. $\mathbf{1}$ is an eigen vector with eigen value equal to 0
3. Symmetric and positive semi-definite
4. Has N non-negative real-valued eigenvalues
5. If the graph ($\mathbf{W}$) has K connected components, then $\mathbf{L}$ has K eigenvectors spanned by $\mathbf{1}_{\mathbf{A_1}}, \ldots, \mathbf{1}_{\mathbf{A_K}}$ with 0 eigenvalue.

Spectral Clustering Algorithm

Observation

- In practice, $\mathbf{W}$ might not have K exactly isolated connected components
- By *perturbation theory*, the smallest eigenvectors of $\mathbf{L}$ will be close to the ideal indicator functions

Algorithm

- Compute first (smallest) K eigen vectors of $\mathbf{L}$
- Let $\mathbf{U}$ be the $N \times K$ matrix with eigenvectors as the columns
- Perform kMeans clustering on the rows of $\mathbf{U}$

References