CSE/STAT 416
Other Clustering Methods
Pre-Class Videos

Tanmay Shah
Paul G. Allen School of Computer Science & Engineering
University of Washington

July 31, 2024
Pre-Class Video 1

Clustering Recap
Define Clusters

In their simplest form, a cluster is defined by

- The location of its center (centroid)
- Shape and size of its spread

**Clustering** is the process of finding these clusters and assigning each example to a particular cluster.

- $x_i$ gets assigned $z_i \in [1, 2, \ldots, k]$
- Usually based on closest centroid

Will define some kind of score for a clustering that determines how good the assignments are

- Based on distance of assigned examples to each cluster
There are many clusters that are harder to learn with this setup

- Distance does not determine clusters
Step 0

Start by choosing the initial cluster centroids

- A common default choice is to choose centroids at random
- Will see later that there are smarter ways of initializing
Step 1

Assign each example to its closest cluster centroid

\[ z_i \leftarrow \arg\min_{j \in [k]} \| \mu_j - x_i \|^2 \]
Update the centroids to be the mean of all the points assigned to that cluster.

\[ \mu_j \leftarrow \frac{1}{n_j} \sum_{i:z_i=j} x_i \]

Computes center of mass for cluster!
Smart Initializing w/ \textit{k-means++}

Making sure the initialized centroids are “good” is critical to finding quality local optima. Our purely random approach was wasteful since it’s very possible that initial centroids start close together.

Idea: Try to select a set of points farther away from each other.

\textbf{k-means++} does a slightly smarter random initialization

1. Choose first cluster \(\mu_1\) from the data uniformly at random
2. For the current set of centroids (starting with just \(\mu_1\)), compute the distance between each datapoint and its closest centroid
3. Choose a new centroid from the remaining data points with probability of \(x_i\) being chosen proportional to \(d(x_i)^2\)
4. Repeat 2 and 3 until we have selected \(k\) centroids
Problems with k-means

In real life, cluster assignments are not always clear cut

Because we minimize Euclidean distance, k-means assumes all the clusters are spherical

We can change this with weighted Euclidean distance
  ▪ Still assumes every cluster is the same shape/orientation
Failure Modes of k-means

If we don’t meet the assumption of spherical clusters, we will get unexpected results.

- Disparate cluster sizes
- Overlapping clusters
- Different shaped/oriented clusters
A much more flexible approach is modeling with a mixture model. Model each cluster as a different probability distribution and learn their parameters:

- E.g. Mixture of Gaussians
- Allows for different cluster shapes and sizes
- Typically learned using Expectation Maximization (EM) algorithm

Allows soft assignments to clusters:

- 54% chance document is about world news, 45% science, 1% conspiracy theory, 0% other
Pre-Class Video 2

Divisive Clustering
Hierarchical Clustering
Lots of data is hierarchical by nature

https://www.techleer.com/articles/478-insight-into-hierarchical-representations-through-poincare-embedding/
Motivation

If we try to learn clusters in hierarchies, we can

- Avoid choosing the # of clusters beforehand
- Use dendrograms to help visualize different granularities of clusters
- Allow us to use any distance metric
  - K-means requires Euclidean distance
- Can often find more complex shapes than k-means
Finding Shapes

k-means

Mixture Models

Hierarchical Clustering
Types of Algorithms

**Divisive, a.k.a. top-down**
- Start with all the data in one big cluster and then recursively split the data into smaller clusters
  - Example: **recursive k-means**

**Agglomerative, a.k.a. bottom-up:**
- Start with each data point in its own cluster. Merge clusters until all points are in one big cluster.
  - Example: **single linkage clustering**
Divisive Clustering

Start with all the data in one cluster, and then repeatedly run k-means to divide the data into smaller clusters. Repeatedly run k-means on each cluster to make sub-clusters.
Using Wikipedia

Wikipedia

Athletes

Non-athletes

Baseball

Soccer/Ice hockey

Musicians, artists, actors

Scholars, politicians, government officials
For divisive clustering, you need to make the following choices:

- Which algorithm to use (e.g., k-means)
- How many clusters per split
- When to split vs when to stop
  - **Max cluster size**
    Number of points in cluster falls below threshold
  - **Max cluster radius**
    distance to furthest point falls below threshold
  - **Specified # of clusters**
    split until pre-specified # of clusters is reached
CSE/STAT 416

Other Clustering Methods

Tanmay Shah
Paul G. Allen School of Computer Science & Engineering
University of Washington

July 31, 2024

❓ Questions? Raise hand or slido #cs416
🎵 Listening to: Still Woozy
Define Clusters

In their simplest form, a cluster is defined by

- The location of its center (centroid)
- Shape and size of its spread

Clustering is the process of finding these clusters and assigning each example to a particular cluster.

- $x_i$ gets assigned $z_i \in [1, 2, \ldots, k]$
- Usually based on closest centroid

Will define some kind of objective function for a clustering that determines how good the assignments are

- Based on distance of assigned examples to each cluster.
- Close distance reflects strong similarity between datapoints.
There are many clusters that are harder to learn with this setup
- Distance does not determine clusters
Visualizing k-means

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
What convergence guarantees do you think we will have with k-means, given a sufficiently large number of iterations?

- Converges to the global optimum
- Converges to a local optima
- None
Smart Initializing w/ \( k \)-means++

Making sure the initialized centroids are “good” is critical to finding quality local optima. Our purely random approach was wasteful since it’s very possible that initial centroids start close together.

Idea: Try to select a set of points farther away from each other.

\( k \)-means++ does a slightly smarter random initialization

1. Choose first cluster \( \mu_1 \) from the data uniformly at random
2. For each datapoint \( x_i \), compute the distance between \( x_i \) and the closest centroid from the current set of centroids (starting with just \( \mu_i \)). Denote that distance \( d(x_i) \).
3. Choose a new centroid from the remaining data points, where the probability of \( x_i \) being chosen is proportional to \( d(x_i)^2 \).
4. Repeat 2 and 3 until we have selected \( k \) centroids.
Assessing Performance
Don’t know, there is no “right answer” in clustering 🤷. Depends on the practitioner’s domain-specific knowledge and interpretation of results!
Which clustering does k-means prefer?

k-means is trying to optimize the **heterogeneity** objective

\[
\arg\min_{z, \mu} \sum_{j=1}^{k} \sum_{i=1}^{n} 1\{z_i = j\} \left\| \mu_j - x_i \right\|_2^2
\]
Coordinate Descent

k-means is trying to minimize the heterogeneity objective

\[
\arg\min_{z,\mu} \sum_{j=1}^{k} \sum_{i=1}^{n} 1\{z_i = j\} \|\mu_j - x_i\|_2^2
\]

Step 0: Initialize cluster centers

Repeat until convergence:

   Step 1: Assign each example to its closest cluster centroid

   Step 2: Update the centroids to be the mean of all the points assigned to that cluster

**Coordinate Descent** alternates how it updates parameters to find minima. On each of iteration of Step 1 and Step 2, heterogeneity decreases or stays the same.

=> Will converge in finite time
Consider training k-means to convergence for different values of k. Which of the following graphs shows how the heterogeneity objective will change based on the value of k?
Consider training k-means to convergence for different values of k. Which of the following graphs shows how the heterogeneity objective will change based on the value of k?

A. 
B. 
C. 
D.
How to Choose k?

No right answer! Depends on your application.

- General, look for the “elbow” in the graph

Note: You will usually have to run k-means multiple times for each k
Cluster shape

- k-means works well for well-separated hyper-spherical clusters of the same size
Clustering vs Classification

- Clustering looks like we assigned labels (by coloring or numbering different groups) but we didn’t use any labeled data.
- In clustering, the “labels” don’t have meaning. To give meaning to the labels, human inputs is required.
- Classification learns from minimizing the error between a prediction and an actual label.
- Clustering learns by minimizing the distance between points in a cluster.
- Classification quality metrics (accuracy / loss) do not apply to clustering (since there is no label).
- You can’t use validation set / cross-validation to choose the best choice of k for clustering.
Recap

- Differences between classification and clustering
- What types of clusters can be formed by k-means
- K-means algorithm
- Convergence of k-means
- How to choose k
- Better initialization using k-means++
Failure Modes of k-means

If we don’t meet the assumption of spherical clusters, we will get unexpected results.

- Disparate cluster sizes
- Overlapping clusters
- Different shaped/oriented clusters
Visualizing Gaussian Mixture Models
Lots of data is hierarchical by nature

Nouns

https://www.techleer.com/articles/478-insight-into-hierarchical-representations-through-poincare-embedding/
Types of Algorithms

**Divisive**, a.k.a. *top-down*
- Start with all the data in one big cluster and then recursively split the data into smaller clusters
  - Example: *recursive k-means*

**Agglomerative**, a.k.a. *bottom-up:*
- Start with each data point in its own cluster. Merge clusters until all points are in one big cluster.
  - Example: *single linkage clustering*
Divisive Clustering

Start with all the data in one cluster, and then repeatedly run k-means to divide the data into smaller clusters. Repeatedly run k-means on each cluster to make sub-clusters.
Using Wikipedia

Athletes

Wikipedia

Baseball

Soccer/Ice hockey

Non-athletes

Scholars, politicians, government officials

Wikipedia

Musicians, artists, actors
Bisecting K-Means
You want to detect outliers in a dataset (shown below).
- How would you use k-means clustering to detect outliers?
- How would you use divisive clustering to detect outliers?
You want to detect outliers in a dataset (shown below).
- How would you use k-means clustering to detect outliers?
- How would you use divisive clustering to detect outliers?
Agglomerative Clustering
Agglomerative Clustering
Agglomerative Clustering

Algorithm at a glance

1. Initialize each point in its own cluster
2. Define a distance metric between clusters

While there is more than one cluster

3. Merge the two closest clusters
Step 1

1. Initialize each point to be its own cluster
2. Define a distance metric between clusters

Single Linkage

$$distance(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$

This formula means we are defining the distance between two clusters as the smallest distance between any pair of points between the clusters.
Step 3

Merge closest pair of clusters
Repeat step 3 until all points are in one cluster.
Repeat

Notice that the height of the dendrogram is growing as we group points farther from each other.
Looking at the dendrogram, we can see there is a bit of an outlier!

Can tell by seeing a point join a cluster with a really large distance.

Repeat step 3 until all points are in one cluster.
Repeat

The tall links in the dendrogram show us we are merging clusters that are far away from each other.

4. Repeat step 3 until all points are in one cluster.
Repeat

Final result after merging all clusters
In what order will the following points get merged into clusters? Use L2 (Euclidean) distance, and the single linkage function.
With agglomerative clustering, we are now very able to learn weirder clusterings like
Dendrogram

x-axis shows the datapoints (arranged in a very particular order)
y-axis shows distance between merged clusters
Dendrogram

The path shows you all clusters that a single point belongs and the order in which its clusters merged.
Choose a distance $D^*$ to “cut” the dendrogram

- Use the largest clusters with distance $< D^*$
- Usually ignore the idea of the nested clusters after cutting
How many clusters would we have if we use this threshold?
How many clusters would we have if we use this threshold?
Cut Dendrogram

Every branch that crosses $D^*$ becomes its own cluster.
For agglomerative clustering, you need to make the following choices:

- Distance metric $d(x_i, x_j)$
- Linkage function
  - Single Linkage:
    $$D(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$
  - Complete Linkage:
    $$D(C_1, C_2) = \max_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$
  - Centroid Linkage
    $$D(C_1, C_2) = d(\mu_1, \mu_2)$$
  - Others
- Where and how to cut dendrogram
Linkage Functions
For visualization, generally a smaller # of clusters is better

For tasks like outlier detection, cut based on:
- Distance threshold
- Or some other metric that tries to measure how big the distance increased after a merge

No matter what metric or what threshold you use, no method is “incorrect”. Some are just more useful than others.
Computational Cost of Agglomerative Clustering

Computing all pairs of distances is pretty expensive!
- A simple implementation takes $O(n^2 \log(n))$

Can be much implemented more cleverly by taking advantage of the **triangle inequality**
- “Any side of a triangle must be less than the sum of its sides”

Best known algorithm is $O(n^2)$
k-means vs. Agglomerative Clustering

- K-means is more efficient on big data than hierarchical clustering.
- Initialization changes results in k-means, not in agglomerative clustering has reproducible results.
- K-means works well only for hyper-spherical clusters, agglomerative clustering can handle more complex cluster shapes.
- K-means requires selecting a number of clusters beforehand. In agglomerative clustering, you can decide on the number of clusters afterwards using the dendrogram.
This week we want to practice recalling vocabulary. Spend 10 minutes trying to write down all the terms for concepts we have learned in this class and try to bucket them into the following categories.

Regression
Classification
Deep Learning
Document Retrieval
Misc – For things that fit in multiple places or none of the above

You don’t need to define/explain the terms for this exercise, but you should know what they are!

Try to do this for at least 5 minutes from recall before looking at your notes!
Recap

- Problems with k-means
- Mixture Models
- Hierarchical clustering
- Divisive Clustering
- Agglomerative Clustering
- Dendrograms