94-775/95-865 Lecture 5: Clustering Part I

George Chen
Announcements

• HW1 solutions are up in Canvas (look in “Files”)

• Sample final project slide decks from last year are up in Canvas (look in “Files”)

• The final project proposal is now due Friday February 15

• Hard part initially: finding data that relates to a policy question you care about!
Let’s look at images
(Flashback) Recap: Basic Text Analysis

- Represent text in terms of “features” (such as how often each word/phrase appears)
- Can repeat this for different documents: represent each document as a “feature vector”

"Sentence": ☀️气象☀️ 伞☂️ ☂️ ☀️

In general (not just text): first represent data as feature vectors

This is a point in 4-dimensional space, \( \mathbb{R}^4 \)

# dimensions = number of terms
Example: Representing an Image

Go row by row and look at pixel values

Image source: starwars.com
Example: Representing an Image

Go row by row and look at pixel values

Image source: starwars.com
Example: Representing an Image

Go row by row and look at pixel values

Image source: starwars.com
Example: Representing an Image

Go row by row and look at pixel values

# dimensions = image width \times image height

Very high dimensional!

Image source: starwars.com
Dimensionality Reduction for Images

Demo
**Visualization is a way of debugging data analysis!**

Example: Trying to understand how people interact in a social network.

**Important:**
Handwritten digit demo is a **toy example** where we know which images correspond to digits 0, 1, … 9.

**Many real UDA problems:**
The data are **messy** and it’s not obvious what the “correct” labels/answers look like, and “correct” is ambiguous!

This is largely why I am covering “supervised” methods (require labels) **after** “unsupervised” methods (don’t require labels).

Top right image source: https://bost.ocks.org/mike/miserables/
Let’s look at a *structured* dataset (easier to explain clustering): drug consumption data
Drug Consumption Data

Demo
Clustering Shows Up Often in Real Data!

- Example: crime might happen more often in specific hot spots

- Example: people applying for micro loans have a few specific uses in mind (education, electricity, healthcare, etc)

- Example: users in a recommendation system can share similar taste in products

- Example: students have different skill levels (clusters could correspond to different letter grades)

To come up with clusters, we first need to define what it means for two things to be “similar”
Defining Similarity

• There usually is no “best” way to define similarity

Example: cosine similarity

\[
\frac{\langle Y_u, Y_v \rangle}{\|Y_u\| \|Y_v\|}
\]

• Also popular: define a distance first and then turn it into a similarity

Example: Euclidean distance

\[
\|Y_u - Y_v\|
\]

Turn into similarity with decaying exponential

\[
\exp(-\gamma \|Y_u - Y_v\|)
\]

where \(\gamma > 0\)
Example: Time Series

How would you compute a distance between these?

Only observe time steps between 0 and $T$
Example: Time Series

How would you compute a distance between these?

$Y_u$

$Y_v$

Only observe time steps between 0 and $T$
Example: Time Series

How would you compute a distance between these?

$Y_W Y_U$

Distance could be defined as the area of this purple shaded in region

One solution: Align them first

In practice: for time series, very popular to use "dynamic time warping" to first align (it works kind of like how spell check does for words)
Is a Similarity Function Any Good?

Easy thing to check:

• Pick a data point

• Compute its similarity to all the other data points, and sort them from most similar to least similar

• Inspect the most similar data points

If the most similar points are not interpretable, it's quite likely that your similarity function isn't very good  =(  

Going from Similarities to Clusters

There's a whole zoo of clustering methods

Two main categories we'll talk about:

**Generative models**
1. Pretend data generated by specific model with parameters
2. Learn the parameters ("fit model to data")
3. Use fitted model to determine cluster assignments

**Hierarchical clustering**
Top-down: Start with everything in 1 cluster and decide on how to recursively split
Bottom-up: Start with everything in its own cluster and decide on how to iteratively merge clusters

We start here
We're going to start with perhaps the most famous of clustering methods. It won't yet be apparent what this method has to do with generative models.
The diagram illustrates the steps of the k-means clustering algorithm.

1. **Step 0: Pick $k$**
   - We’ll pick $k = 2$.

2. **Step 1: Pick guesses for where cluster centers are**
   - Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers.
Step 0: Pick $k$

We’ll pick $k = 2$

$k$-means

Step 1: Pick guesses for where cluster centers are

Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers
(There are many ways to make the initial guesses)
**k-means**

**Step 0: Pick \( k \)**
- We’ll pick \( k = 2 \)

**Step 1: Pick guesses for where cluster centers are**
- Example: choose \( k \) of the points uniformly at random to be initial guesses for cluster centers
- (There are many ways to make the initial guesses)

**Step 2: Assign each point to belong to the closest cluster**
**Step 0:** Pick $k$

We’ll pick $k = 2$

**Step 1:** Pick guesses for where cluster centers are

Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers

(There are many ways to make the initial guesses)

**Step 2:** Assign each point to belong to the closest cluster

**Step 3:** Update cluster means (to be the center of mass per cluster)
**k-means**

**Step 0: Pick k**
We’ll pick $k = 2$

**Step 1: Pick guesses for where cluster centers are**
Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers
(There are many ways to make the initial guesses)

**Step 2: Assign each point to belong to the closest cluster**

**Step 3: Update cluster means** (to be the center of mass per cluster)
**Step 0: Pick $k$**

We’ll pick $k = 2$

**Step 1: Pick guesses for where cluster centers are**

Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers

(There are many ways to make the initial guesses)

**Repeat**

Step 2: Assign each point to belong to the closest cluster

Step 3: Update cluster means (to be the center of mass per cluster)
**k-means**

**Step 0: Pick $k$**

We’ll pick $k = 2$

**Step 1: Pick guesses for where cluster centers are**

Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers.

(There are many ways to make the initial guesses.)

**Step 2: Assign each point to belong to the closest cluster**

**Step 3: Update cluster means (to be the center of mass per cluster)**

Repeat
**k-means**

**Step 0: Pick k**
We’ll pick $k = 2$

**Step 1: Pick guesses for where cluster centers are**
Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers
(There are many ways to make the initial guesses)

**Step 2: Assign each point to belong to the closest cluster**

**Step 3: Update cluster means (to be the center of mass per cluster)**

Repeat
**k-means**

**Step 0: Pick** \( k \)

We’ll pick \( k = 2 \)

**Step 1: Pick guesses for where cluster centers are**

Example: choose \( k \) of the points uniformly at random to be initial guesses for cluster centers

(There are many ways to make the initial guesses)

**Repeat**

**Step 2: Assign each point to belong to the closest cluster**

**Step 3: Update cluster means** (to be the center of mass per cluster)
**k-means**

**Step 0:** Pick $k$

We’ll pick $k = 2$

**Step 1:** Pick guesses for where cluster centers are

Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers

(There are many ways to make the initial guesses)

**Step 2:** Assign each point to belong to the closest cluster

**Step 3:** Update cluster means (to be the center of mass per cluster)

**Repeat**
**$k$-means**

**Step 0: Pick $k$**
We’ll pick $k = 2$

**Step 1: Pick guesses for where cluster centers are**
Example: choose $k$ of the points uniformly at random to be initial guesses for cluster centers
(There are many ways to make the initial guesses)

Repeat until convergence:

**Step 2: Assign each point to belong to the closest cluster**

**Step 3: Update cluster means (to be the center of mass per cluster)**
**$k$-means**

Final output: cluster centers, cluster assignment for every point

Remark: Very sensitive to choice of $k$ and initial cluster centers

How to pick $k$?
- Basic check: If you have really, really tiny clusters $\Rightarrow$ decrease $k$
- More details later

Suggested way to pick initial cluster centers: “$k$-means++” method
(rough intuition: incrementally add centers; favor adding center far away from centers chosen so far)
When does $k$-means work well?

$k$-means is related to a more general model, which will help us understand $k$-means.
Gaussian Mixture Model (GMM)

What random process could have generated these points?
Generative Process

Think of flipping a coin

each outcome: heads or tails

Each flip doesn't depend on any of the previous flips
Generative Process

Think of flipping a coin
each outcome: 2D point

Each flip doesn't depend on any of the previous flips

*Okay, maybe it's bizarre to think of it as a coin…*

*If it helps, just think of it as you pushing a button and a random 2D point appears…*
Gaussian Mixture Model (GMM)

We now discuss a way to generate points in this manner.
Gaussian Mixture Model (GMM)

Assume: points sampled independently from a probability distribution

This is the sum of two 2D Gaussian distributions!

how probable point generated at \((x, y)\) is

Red = more likely
Blue = less likely

Example of a 2D probability distribution

Image source: https://www.intechopen.com/source/html/17742/media/image25.png
Quick Reminder: 1D Gaussian

This is a 1D Gaussian distribution

Image source: https://matthew-brett.github.io/teaching//smoothing_intro-3.hires.png
2D Gaussian

This is a 2D Gaussian distribution

Image source: https://i.stack.imgur.com/OLWce.png
Gaussian Mixture Model (GMM)

Assume: points sampled independently from a probability distribution

This is the sum of two 2D Gaussian distributions!

Key idea: Each Gaussian corresponds to a different cluster

Red = more likely
Blue = less likely

Example of a 2D probability distribution

Image source: https://www.intechopen.com/source/html/17742/media/image25.png
Gaussian Mixture Model (GMM)

- For a fixed value $k$ and dimension $d$, a GMM is the sum of $k$ $d$-dimensional Gaussian distributions so that the overall probability distribution looks like $k$ mountains (We've been looking at $d = 2$)
  - Each mountain corresponds to a different cluster
  - Different mountains can have different peak heights
  - One missing thing we haven't discussed yet: different mountains can have different shapes
2D Gaussian Shape

In 1D, you can have a skinny Gaussian or a wide Gaussian

Less uncertainty  More uncertainty

In 2D, you can more generally have ellipse-shaped Gaussians

Ellipse enables encoding relationship between variables  Can't have arbitrary shapes

Top-down view of an example 2D Gaussian distribution

Image source: https://www.cs.colorado.edu/~mozer/Teaching/syllabi/ProbabilisticModels2013/homework/assign5/a52dgauss.jpg
Gaussian Mixture Model (GMM)

- For a fixed value $k$ and dimension $d$, a GMM is the sum of $k$ $d$-dimensional Gaussian distributions so that the overall probability distribution looks like $k$ mountains (We've been looking at $d = 2$)
  - Each mountain corresponds to a different cluster
  - Different mountains can have different peak heights
  - Different mountains can have different ellipse shapes (captures "covariance" information)
Example: 1D GMM with 2 Clusters

Cluster 1

Probability of generating a point from cluster 1 = 0.5
Gaussian mean = −5
Gaussian std dev = 1

Cluster 2

Probability of generating a point from cluster 2 = 0.5
Gaussian mean = 5
Gaussian std dev = 1

What do you think this looks like?
Example: 1D GMM with 2 Clusters

Cluster 1

Probability of generating a point from cluster 1 = 0.5
Gaussian mean = −5
Gaussian std dev = 1

Cluster 2

Probability of generating a point from cluster 2 = 0.5
Gaussian mean = 5
Gaussian std dev = 1
**Example: 1D GMM with 2 Clusters**

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>Cluster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of generating a point from cluster 1 = 0.7</td>
<td>Probability of generating a point from cluster 2 = 0.3</td>
</tr>
<tr>
<td>Gaussian mean = −5</td>
<td>Gaussian mean = 5</td>
</tr>
<tr>
<td>Gaussian std dev = 1</td>
<td>Gaussian std dev = 1</td>
</tr>
</tbody>
</table>

What do you think this looks like?
Example: 1D GMM with 2 Clusters

Cluster 1

Probability of generating a point from cluster 1 = 0.7
Gaussian mean = $-5$
Gaussian std dev = 1

Cluster 2

Probability of generating a point from cluster 2 = 0.3
Gaussian mean = 5
Gaussian std dev = 1
### Example: 1D GMM with 2 Clusters

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>Cluster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of generating a point from cluster 1 = 0.7</td>
<td>Probability of generating a point from cluster 2 = 0.3</td>
</tr>
<tr>
<td>Gaussian mean = -5</td>
<td>Gaussian mean = 5</td>
</tr>
<tr>
<td>Gaussian std dev = 1</td>
<td>Gaussian std dev = 1</td>
</tr>
</tbody>
</table>

How to generate 1D points from this GMM:

1. Flip biased coin (with probability of heads 0.7)
2. If heads: sample 1 point from Gaussian mean -5, std dev 1
   If tails: sample 1 point from Gaussian mean 5, std dev 1
Example: 1D GMM with 2 Clusters

**Cluster 1**
- Probability of generating a point from cluster 1 = $\pi_1$
- Gaussian mean = $\mu_1$
- Gaussian std dev = $\sigma_1$

**Cluster 2**
- Probability of generating a point from cluster 2 = $\pi_2$
- Gaussian mean = $\mu_2$
- Gaussian std dev = $\sigma_2$

How to generate 1D points from this GMM:
1. Flip biased coin (with probability of heads $\pi_1$)
2. If heads: sample 1 point from Gaussian mean $\mu_1$, std dev $\sigma_1$
   - If tails: sample 1 point from Gaussian mean $\mu_2$, std dev $\sigma_2$
Example: 1D GMM with $k$ Clusters

Cluster 1

Probability of generating a point from cluster 1 = $\pi_1$
Gaussian mean = $\mu_1$
Gaussian std dev = $\sigma_1$

Cluster $k$

Probability of generating a point from cluster $k = \pi_k$
Gaussian mean = $\mu_k$
Gaussian std dev = $\sigma_k$

How to generate 1D points from this GMM:

1. Flip biased $k$-sided coin (the sides have probabilities $\pi_1, \ldots, \pi_k$)
2. Let $Z$ be the side that we got (it is some value 1, \ldots, $k$)
3. Sample 1 point from Gaussian mean $\mu_Z$, std dev $\sigma_Z$
Example: 1D GMM with $k$ Clusters

Cluster 1

Probability of generating a point from cluster 1 $= \pi_1$
Gaussian mean $= \mu_1$
Gaussian std dev $= \sigma_1$

Cluster $k$

Probability of generating a point from cluster $k = \pi_k$
Gaussian mean $= \mu_k$
Gaussian std dev $= \sigma_k$

How to generate 1D points from this GMM:
1. Flip biased $k$-sided coin (the sides have probabilities $\pi_1, \ldots, \pi_k$)
2. Let $Z$ be the side that we got (it is some value $1, \ldots, k$)
3. Sample 1 point from Gaussian mean $\mu_Z$, std dev $\sigma_Z$
Example: 2D GMM with $k$ Clusters

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>Cluster $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of generating a point from cluster 1 = $\pi_1$</td>
<td>Probability of generating a point from cluster $k = \pi_k$</td>
</tr>
<tr>
<td>Gaussian mean = $\mu_1$ 2D point</td>
<td>Gaussian mean = $\mu_k$ 2D point</td>
</tr>
<tr>
<td>Gaussian covariance = $\Sigma_1$ 2x2 matrix</td>
<td>Gaussian covariance = $\Sigma_k$ 2x2 matrix</td>
</tr>
</tbody>
</table>

How to generate 2D points from this GMM:
1. Flip biased $k$-sided coin (the sides have probabilities $\pi_1$, $\ldots$, $\pi_k$)
2. Let $Z$ be the side that we got (it is some value 1, $\ldots$, $k$)
3. Sample 1 point from Gaussian mean $\mu_Z$, covariance $\Sigma_Z$
How to generate points from this GMM:
1. Flip biased $k$-sided coin (the sides have probabilities $\pi_1, \ldots, \pi_k$)
2. Let $Z$ be the side that we got (it is some value $1, \ldots, k$)
3. Sample 1 point from Gaussian mean $\mu_Z$, covariance $\Sigma_Z$
High-Level Idea of GMM

• Generative model that gives a *hypothesized* way in which data points are generated

  In reality, data are unlikely generated the same way!

  In reality, data points might not even be independent!
“All models are wrong, but some are useful.”

–George Edward Pelham Box
High-Level Idea of GMM

• Generative model that gives a hypothesized way in which data points are generated

  In reality, data are unlikely generated the same way!

  In reality, data points might not even be independent!

• Learning ("fitting") the parameters of a GMM
  • Input: $d$-dimensional data points, your guess for $k$
  • Output: $\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k$

• After learning a GMM:
  • For any $d$-dimensional data point, can figure out probability of it belonging to each of the clusters

  How do you turn this into a cluster assignment?