Unstructured Data Analysis

Lecture 8: More clustering, topic modeling

George Chen
More on Automatic Selection of $k$

Dirichlet Process Gaussian Mixture Model (DP-GMM):

- Number of clusters is effectively random, and *can grow with the amount of data you have!*

- While you don't have to choose $k$, you have to choose a different parameter which says how large clusters are
DP-GMM High-Level Idea

Cluster 1

Probability of generating a point from cluster 1 = π₁
Gaussian mean = μ₁
Gaussian covariance = Σ₁

Cluster 2

There is a parameter that controls how these π values roughly decay
π₂
π₃

Cluster 3

μ₂
μ₃
Σ₂
Σ₃

It goes on forever!

There are an infinite number of parameters

(Rough idea) How to generate points from this DP-GMM:
1. Flip biased ∞-sided coin (the sides have probabilities π₁, π₂, π₃, …)
2. Let Z be the side that we got (it is a positive integer)
3. Sample 1 point from Gaussian mean μₚ, covariance Σₚ

Remark: For any given dataset, when learning the DP-GMM, there aren't going to be an infinite number of clusters found
More on Automatic Selection of $k$

Dirichlet Process Gaussian Mixture Model (DP-GMM):

- Number of clusters is effectively random, and can grow with the amount of data you have!

- While you don't have to choose $k$, you have to choose a different parameter which says how large clusters are

- An example of a Bayesian nonparametric model (roughly: a generative model with an infinite number of parameters, where the parameters are random)
Learning a DP-GMM

Two main approaches:

- Finite approximation where you specify some maximum number of possible clusters (the algorithm will find up to that many clusters)
  
  This is what’s implemented in sklearn

- Algorithm is somewhat similar to $k$-means/EM for GMMs

- Algorithm output: very similar to regular GMM fitting

- Random sampling approach (no finite approximation needed!)

  - Algorithm output: a bunch of samples of different cluster assignments (can pick one with highest probability)

  This is what’s implemented in R (package dpmixsim)
Learning a DP-GMM

Demo
This next algorithm will give you a sense of how we get around specifying the number of clusters directly.

\[ k\text{-means approximates (a special case of) learning GMM's.} \]

What approximates learning DP-GMMs?

This next algorithm will give you a sense of how we get around specifying the number of clusters directly.
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster
Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

“Step 2a”. Pick point outside of gray coverage to make new cluster
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

“Step 2a”. Pick point outside of gray coverage to make new cluster
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

“Step 2a”. Pick point outside of gray coverage to make new cluster

“Step 2b”. Assign closest points to current clusters
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

“Step 2a”. Pick point outside of gray coverage to make new cluster

“Step 2b”. Assign closest points to current clusters
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

“Step 2a”. Pick point outside of gray coverage to make new cluster

“Step 2b”. Assign closest points to current clusters
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

“Step 2a”. Pick point outside of gray coverage to make new cluster

“Step 2b”. Assign closest points to current clusters
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2a. Pick point outside of gray coverage to make new cluster

Step 2b. Assign closest points to current clusters

Step 3. Recompute cluster centers
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2a. Pick point outside of gray coverage to make new cluster

Step 2b. Assign closest points to current clusters

Step 3. Recompute cluster centers
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2a. Pick point outside of gray coverage to make new cluster

Step 2b. Assign closest points to current clusters

Step 3. Recompute cluster centers
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 3. Recompute cluster centers

“Step 2a”. Pick point outside of gray coverage to make new cluster

“Step 2b”. Assign closest points to current clusters
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2a. Pick point outside of gray coverage to make new cluster

Step 2b. Assign closest points to current clusters

Step 3. Recompute cluster centers
Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2. For each point:
   (a) If it's not currently covered by gray balls, make it a new cluster center
   (b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in the same cluster

Step 2. For each point:
   (a) If it’s not currently covered by gray balls, make it a new cluster center
   (b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2. For each point:
   (a) If it's not currently covered by gray balls, make it a new cluster center
   (b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2. For each point:
   (a) If it’s not currently covered by gray balls, make it a new cluster center
   (b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
DP-means

Step 0. Pick concentration parameter $\lambda > 0$

Step 1. Start with everything in same cluster

Step 2. For each point:

(a) If it’s not currently covered by gray balls, make it a new cluster center

(b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
Step 0: Pick concentration parameter $\lambda > 0$

Step 1: Start with everything in same cluster

Step 2. For each point:
   
   (a) If it’s not currently covered by gray balls, make it a new cluster center
   
   (b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
DP-means

Step 0: Pick concentration parameter $\lambda > 0$

Step 1: Start with everything in same cluster

Step 2. For each point:
   (a) If it's not currently covered by gray balls, make it a new cluster center
   (b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
DP-means

Step 0: Pick concentration parameter $\lambda > 0$

Step 1: Start with everything in same cluster

Step 2. For each point:
(a) If it’s not currently covered by gray balls, make it a new cluster center
(b) Otherwise assign it to nearest cluster

Step 3. Recompute cluster centers
DP-means

Step 0: Pick concentration parameter \( \lambda > 0 \)

Step 1: Start with everything in same cluster

Step 2. For each point:

Step 3. Recompute cluster centers

Repeat until convergence:

(a) If it’s not currently covered by gray balls, make it a new cluster center

(b) Otherwise assign it to nearest cluster
As you saw in the DP-GMM demo (and is similar with DP-means), DP-means can produce a few extra small clusters. In practice, can reassign points in small clusters to bigger clusters.
As you saw in the DP-GMM demo (and is similar with DP-means), DP-means can produce a few extra small clusters.

In practice: can reassign points in small clusters to bigger clusters.
As you saw in the DP-GMM demo (and is similar with DP-means), DP-means can produce a few extra small clusters.

In practice: can reassign points in small clusters to bigger clusters.

Can recompute cluster centers.
DP-means

As you saw in the DP-GMM demo (and is similar with DP-means), DP-means can produce a few extra small clusters.

In practice: can reassign points in small clusters to bigger clusters.
Can recompute cluster centers.
Big picture: DP-means & DP-GMM have a parameter roughly controlling size of clusters rather than number of clusters

If your problem can more naturally be thought of as having cluster sizes that should not be too large, can use DP-means/DP-GMM instead of k-means/GMM

Real example. *Satellite image analysis of rural India to find villages*

Each cluster is a village: don’t know how many villages there are total but rough upper bound on radius of village can be specified

→ DP-means provides a decent solution!
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

I’ll post material on this; won’t be covered in lecture and won’t be on Quiz 2
# Going from Similarities to Clusters

## 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

You learn a model ➔ can predict cluster assignments for points not seen in training

Most popular models assume Euclidean distance…

## 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

Easily works with different distances (not just Euclidean)

Great for problems that don’t need to predict clusters for future points
Example: Hierarchical Clustering on US Counties

(using opioid death rate data across 37 years)

No need to predict which cluster new counties should belong to, since we’re already looking at all U.S. counties!

Image source: Amanda Coston
Clustering Last Remarks

Ultimately, you have to decide on which clustering method and number of clusters make sense for your data.

- After you run a clustering algorithm, make visualizations to interpret the clusters in the context of your application!
- Do not just blindly rely on numerical metrics (e.g., CH index)
- Some times it makes more sense to define your own score function for how good a clustering assignment is.

If you can set up a prediction task, then you can use the prediction task to guide the clustering.
Is clustering structure enough?

- Fundamentally, clustering models assume each point comes from a **single** cluster.
- In reality, a data point could have “mixed” membership and belong to **multiple** clusters.
Topic Modeling

Text

Each document is part of multiple topics
Each topic consists of a bunch of regularly co-occurring words
(example topics: “sports”, “medicine”, “movies”, “finance”)

Movie recommendation

Each user is part of multiple “clusters”/topics
Each cluster/topic consists of a bunch of movies
(example clusters: “sci-fi epics”, “cheesy rom-coms”)

Health care

Each patient’s health records explained by multiple “topics”
Each topic consists of co-occurring “events”
(example topics: “heart condition”, “severe pancreatitis”)
Topic Modeling

Text

Each document is part of multiple topics

Each topic consists of a bunch of regularly co-occurring words
(example topics: “sports”, “medicine”, “movies”, “finance”)

In all of these examples:

• Each data point (a feature vector) is part of multiple topics
• Each topic corresponds to specific feature values in the feature vector likely appearing

Health care

Each patient’s health records explained by multiple “topics”

Each topic consists of co-occurring “events”
(example topics: “heart condition”, “severe pancreatitis”)
Latent Dirichlet Allocation (LDA)

- Easy to describe in terms of text (but works for not just text)
- A generative model
- Input: “document-word” matrix, and pre-specified # topics $k$
- Output: what the $k$ topics are (details on this shortly)

*i*-th row, *j*-th column: # times word *j* appears in doc *i*

Document

\[
\begin{array}{cccc}
1 & 2 & \cdots & d \\
1 & & & \\
2 & & & \\
\vdots & & & \\
n & & & \\
\end{array}
\]
Each word in Alice’s text is generated by:

1. Flip 2-sided coin for Alice
2. If weather: flip 4-sided coin for weather
   If food: flip 4-sided coin for food
Each word in Bob’s text is generated by:

1. Flip 2-sided coin for Bob
2. If weather: flip 4-sided coin for weather
   If food: flip 4-sided coin for food
Each word in doc. $i$ is generated by:

1. Flip 2-sided coin for doc. $i$
2. If weather: flip 4-sided coin for weather
   If food: flip 4-sided coin for food

“Learning the topics” means figuring out these 4-sided coin probabilities
LDA models each word in document \( i \) to be generated as:

- Randomly choose a topic \( Z \) (use topic distribution for doc \( i \))
- Randomly choose a word (use word distribution for topic \( Z \))
LDA

• Easy to describe in terms of text (but works for not just text)
• A generative model
• Input: “document-word” matrix, and pre-specified # topics $k$

$\begin{array}{cccc}
\text{Word} & 1 & 2 & \ldots & d \\
i\text{-th row, } j\text{-th column: # times word } j & 1 & & & \\
2 & & & & \\
\vdots & & & & \\
n & & & & \\
\end{array}$

• Output: the $k$ topics’ distribution of words