95-865 Unstructured Data Analytics

Lecture 8: Clustering (cont’d)

Slides by George H. Chen
Adminstrivia

• Reminder: Your Quiz 1 is this Friday in A301, 5pm-6:20pm
  • In-person, on paper
  • Each quiz is 80 minutes
• No electronics may be used during the exam (e.g., do not use a laptop, tablet, phone, calculator)
  • Open notes (must be on paper and not electronic) — you may bring as much notes as you would like
• Quiz 1 coverage: everything up to and including last Friday's (March 29) recitation
• There is an optional Quiz 1 review session tomorrow (7pm over Zoom)
  • The focus is on last semester's quiz but you are welcome to of course ask questions about other past quizzes & about concepts more generally
(Flashback) Learning a GMM

Step 0: Guess $k$

Step 1: Guess cluster probabilities, means, and covariances

(often done using $k$-means)

Repeat until convergence:

Step 2: Compute probability of each point being in each of the $k$ clusters

Step 3: Update cluster probabilities, means, and covariances accounting for probabilities of each point belonging to each of the clusters

This algorithm is called the **Expectation-Maximization (EM)** algorithm for GMMs (and approximately does maximum likelihood)

(Note: EM by itself is a general algorithm not just for GMMs)
(Rough Intuition) How Shape is Encoded by a GMM

For this ellipse-shaped Gaussian, point B is considered more similar to the cluster center than point A.

\[ k \text{-means would think that point A and point B are equally similar to the cluster center (since both points are distance } r \text{ away from the center).} \]
Relating $k$-means to GMMs

If the ellipses are all circles and have the same "skinniness" (e.g., in the 1D case it means they all have same variance):

- $k$-means approximates the EM algorithm for GMMs (as there is no need to keep track of cluster shape)
- $k$-means does a "hard" assignment of each point to a cluster, whereas the EM algorithm does a "soft" (probabilistic) assignment

Interpretation: When the data appear as if they're from a GMM with true clusters that "look like circles of equal size", then $k$-means should work well
$k$-means should do well on this
But not on this
Relating \( k \)-means to GMMs

If the ellipses are all circles and have the same "skinniness" (e.g., in the 1D case it means they all have same variance):

- \( k \)-means approximates the EM algorithm for GMMs (as there is no need to keep track of cluster shape)

- \( k \)-means does a "hard" assignment of each point to a cluster, whereas the EM algorithm does a "soft" (probabilistic) assignment

**Interpretation:** When the data appear as if they're from a GMM with true clusters that "look like circles of equal size", then \( k \)-means should work well

This is not the only scenario in which \( k \)-means should work well
Even if data aren’t generated from a GMM, $k$-means and GMMs can still cluster correctly
This dataset obviously doesn’t appear to be generated by a GMM $k$-means with $k = 2$, and 2-component GMM will both work well in identifying the two shapes as separate clusters.

Key idea: the clusters are very well-separated (so that many clustering algorithms will work well in this case!)
$k$-means & GMMs, Sketch of Interpretation

Demo
Automatically Choosing the Number of Clusters $k$

For $k = 2, 3, \ldots$ up to some user-specified max value:

- Fit model ($k$-means or GMM) using $k$
- Compute a score for the model
- But what score function should we use?
- Use whichever $k$ has the best score

No single way of choosing $k$ is the “best” way
Here’s an example of a score function you don’t want to use
Residual Sum of Squares

Look at one cluster at a time

Cluster 1

Cluster 2
Residual Sum of Squares

Look at one cluster at a time

Cluster 1

Cluster 2
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center

Cluster 1

Cluster 2
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center

Cluster 1

Cluster 2
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center

Cluster 1

Cluster 2
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center

Cluster 1

Cluster 2

Residual sum of squares for cluster 1: sum of squared purple lengths
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center

Cluster 1

Cluster 2

Residual sum of squares for cluster 1:

$$\text{RSS}_1 = \sum_{x \in \text{cluster 1}} \left\| x - \mu_1 \right\|^2$$
Residual Sum of Squares

Look at one cluster at a time

Measure distance from each point to its cluster center

Repeat similar calculation for other cluster

Residual sum of squares for cluster 2:

$$\text{RSS}_2 = \sum_{x \in \text{cluster 2}} \|x - \mu_2\|^2$$
Residual Sum of Squares

\[ \text{RSS} = \text{RSS}_1 + \text{RSS}_2 = \sum_{x \in \text{cluster 1}} \| x - \mu_1 \|^2 + \sum_{x \in \text{cluster 2}} \| x - \mu_2 \|^2 \]

In general if there are \( k \) clusters:

\[ \text{RSS} = \sum_{g=1}^{k} \text{RSS}_g = \sum_{g=1}^{k} \sum_{x \in \text{cluster } g} \| x - \mu_g \|^2 \]

Remark: \( k \)-means tries to minimize RSS for a fixed value of \( k \) (it does so \textit{approximately}, with no guarantee of optimality)

RSS does not account for clusters having, for instance, ellipse shapes.
Why is minimizing RSS a bad way to choose $k$?

What happens when $k$ is equal to the number of data points?
A Good Way to Choose $k$

RSS measures *within-cluster variation*

$$W = \text{RSS} = \sum_{g=1}^{k} \text{RSS}_g = \sum_{g=1}^{k} \sum_{x \in \text{cluster } g} \|x - \mu_g\|^2$$

Want to also measure *between-cluster variation*

$$B = \sum_{g=1}^{k} (\# \text{ points in cluster } g) \|\mu_g - \mu\|^2$$

Called the CH index

[Calinski and Harabasz 1974]

A good score function to use for choosing $k$:

$$\text{CH}(k) = \frac{B \cdot (n - k)}{W \cdot (k - 1)}$$

$n = \text{total } \# \text{ points}$

Pick $k$ with highest $\text{CH}(k)$

(Choose $k$: among 2, 3, … up to pre-specified max)
Automatically Choosing the Number of Clusters $k$

Demo
What about unstructured data?
Clustering on Images

Demo
Clustering on Text

Basic visualization strategy

Compute most probable words (just like in Lecture 2 using raw counts)

Cluster 1
- Top words for cluster 1

Cluster 2
- Top words for cluster 2

→ We can then compare top words across clusters

Per cluster, can compute other information aside from top words (such as the distribution of document lengths per cluster, or some co-occurrence information per cluster—an example of these ideas is in the Spring 2023 Quiz 1)
Last Remarks on Clustering

• We only saw two clustering methods ($k$-means, GMM)
• We only saw one general strategy to automatically choose # of clusters
  • You must specify a score function — no score function is perfect
• There are lots of clustering methods out there!
  • Many do not require specifying # of clusters (DP-means, DP-GMM, many variants of hierarchical clustering, DBSCAN, OPTICS, …)
• 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)