NUS | Computing
National University of Singapore

Eric Han
eric_han@nus.edu.sg
https://eric-han.com
Computer Science

T10 – 14 Nov 2024

Week 13

CS2109s TG35,36

Content I

Student Feedback on Teaching (SFT)

NUS Student Feedback https://blue.nus.edu.sg/blue/:

- Don't Mix module/grading/project feedback - **feedback only for teaching**.
- Feedback is confidential to university and anonymous to us.
- Feedback is optional but highly encouraged.
- Past student feedback improves teaching; see https://www.eric-han.com/teaching
  - ie. Telegram access, More interactivity.
- Your feedback is important to me, and will be used to improve my teaching.
  - Good $>$ Positive feedback $>$ Encouragement
    - Teaching Awards (nominate)
    - Steer my career path
  - Bad $>$ Negative feedback (nicely pls) $>$ Learning
    - Improvement
    - Better learning experience

In case you want to go and review some of our bonus questions, Wenzhong from TG04/2324s1 (some differences) has completed them all! With permission from him, he have agreed to share his solutions with all of you:

https://github.com/LWZ19/CS2109s-2324s1-bonus

Section 1: **K-means algorithm**

Algorithm 1: K-means clustering

for $k = 1$ ***to*** K **do**
  $\mu_k \leftarrow$ random location
while *not converged* **do**
  for $i = 1$ ***to*** m **do**
    $c^{(i)} \leftarrow argmin_k ||x^{(i)} - \mu_k||^2$
  for $k = 1$ ***to*** K **do**
    $\mu_k \leftarrow \frac{1}{|\{x^{(i)}|c^{(i)}=k\}|} \sum_{x \in \{x^{(i)}|c^{(i)}=k\}} x$

Recap

1. What is the key idea between K-means?

Question 1 [G]

Prove that the algorithm...

i. always produces a partition with a lower loss (monotonically decreasing)
ii. always converges [1].
iii. [@] What is EM algorithm and does it relate to K-Means?

Fun Fact: K-Means is my first ML algorithm that I implemented.

[1] centroids/medoids do not change after an iteration

Answer

- a. Always produces a partition with a lower or eq loss...
  - a. Fix assignment, find the mean points $(|a+b|^2 = |a|^2 + |b|^2 + 2 < a, b >)$
  - b. Fix mean point, find the new assignment. (By definition of L5)
- b. Always converges... (pigeonhole principle + loss never increases)
  - a. There are k^N possible config to partition N data points into k clusters
  - b. So we are transiting from one config to the next.
  - c. The next config has lower or eq loss
  - d. There cannot be a cycle where the next is always lower.
  - e. So must converge in finite number of iterations.

Question 2

Although k-means always converges, it may get stuck at a bad local minimum. What are some ways to help?

Recap

1. Run the algorithm multiple times, what happens?

Answer

The issue is with the initalization:

1. Choose the first centroid randomly then the next to be as far as possible from the first, etc...
2. K-means++, first centroid randomly and choose the rest using some probability distribution.

Question 3

i	1	2	3	4	5	6
x	1	1	2	2	3	3
y	0	1	1	2	1	2

Table 1: 6 data points on a 2D-plane

Cluster the 6 points in table 1 into **two** clusters using the K-means algorithm. The two initial centroids are (0, 1) and (2.5, 2).

Answer

Iteration 1

Using first centroid = (0, 1) and second centroid = (2.5, 2), we get the table below.

Point	D^2 to first centroid	D^2 to second centroid	Assigned Cluster
1	2	6.25	1
2	1	3.25	1
3	4	1.25	2
4	5	0.25	2
5	9	1.25	2
6	10	0.25	2

Computing the new centroids:

- Centroid 1 = ((1, 0) + (1, 1)) / 2 = (1, 0.5)
- Centroid 2 = ((2, 1) + (2, 2) + (3, 1) + (3, 2)) / 4 = (2.5, 1.5)

Iteration 2

Using first centroid = (1, 0.5) and second centroid = (2.5, 1.5), we get the table below.

Point	D^2 to first centroid	D^2 to second centroid	Assigned Cluster
1	0.25	4.5	1
2	0.25	2.5	1
3	1.25	0.5	2
4	3.25	0.5	2
5	4.25	0.5	2
6	6.25	0.5	2

Computing the new centroids:

- Centroid 1 = ((1, 0) + (1, 1)) / 2 = (1, 0.5)
- Centroid 2 = ((2, 1) + (2, 2) + (3, 1) + (3, 2)) / 4 = (2.5, 1.5)

Since the centroids are the same as those from the previous iteration, the K-means algorithm has converged.

Question 4

Cluster the 6 points in table 1 into **two** clusters using the K-medoids algorithm. The initial medoids are point 1 and point 3.

Recap

1. What is the key difference between K-means and K-medoids?

Answer

Iteration 1

Point	D^2 to first medoid	D^2 to second medoid	Assigned Cluster
1	0	2	1
2	1	1	2
3	2	0	2
4	5	1	2
5	5	1	2
6	8	2	2

For point 2, the distance between itself to the first medoid is the same as the distance between itself to the second medoid.

- For simplicity, we assign point 2 as a member of the second cluster.
- The strategy chosen must be deterministic.

Computing the new medoid:

- Centroid 1 = (1, 0)
- Centroid 2 = ((1, 1) + (2, 1) + (2, 2) + (3, 1) + (3, 2)) / 5 = (2.2, 1.4)

We need to find the closest points to each centroid.

- For centroid 1, the closest point is point 1. Hence, we set point 1 as the new medoid.
- For centroid 2, the closest point is point 3. Hence, we set point 3 as the new medoid.

Since the medoids are the same as the initial ones, the K-medoids algorithm has converged.

Section 2: **Hierarchical clustering**

Question

Given this dataset:

i	1	2	3	4	5
x_1	0	1	3	1	1
x_2	0	1	0	3	4

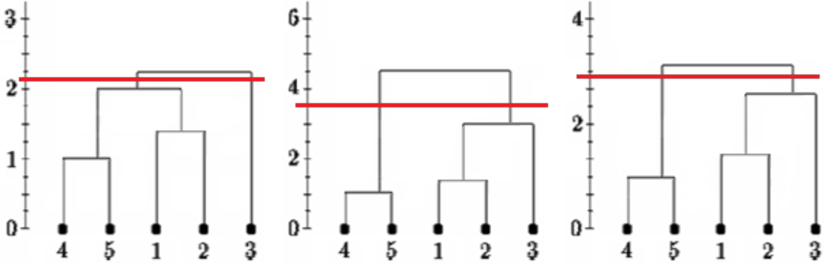
1. Complete the distance matrix (using square of Euclidean distance).
2. Draw the dendrogram for the three linkage methods (Single, Complete and Centroid).
3. Draw a line that partitions it into 2 clusters.

Recap

What is the algorithm to construct a hierarchical cluster?

Answer

	1	2	3	4	5
1	0				
2	2	0			
3	9	5	0		
4	10	4	13	0	
5	17	9	20	1	0



(a) Single linkage (b) Complete linkage (c) Centroid linkage

Section 3: **SVD**

Using the `tut10.ipynb`, we study PCA:

1. The current choice of $k = 9$ does not produce a very nice output. What is a good value for k?
2. For the value of k you select in (a), what is the space saved by doing this compression?
3. What are the drawbacks of this form of compression?
4. [@] How does JPEG work and relate to this technique?
5. [@] Should we mean-center the data? How does the calculation change?
6. [@] What happens when we use the largest k?

Recap

- What is PCA and how does it work?

Answer 1

Figure 2: $k = 286$ gives us 99% of the variance

Answer 2

When using $k = 286$, the (512×1536) 2D-array is now represented by U_{reduce} (512×286) and Z (286×1536). This demonstrates approximately 25.5% space saved.

$$\frac{(512 \times 286) + (286 \times 1536)}{512 \times 1536} = \frac{585728}{786432} = 0.745$$

If we wish to have more compression, we can choose a smaller k, but at the expense of image quality.

Answer 3

- Lossy Compression - The image cannot be reconstructed exactly and permanently loses information - ie. 100% of variance.
- Using the full U_{reduce}, we actually use more space than the original.

Section 4: **Wrapping up**

Parting Advice

- NUS: 2023 GES Employment Rates
- AI Hype comes in cycles.
- Do what interest you - I picked AI/ML because I love it.

Recommended Next Modules (If you like AI?)

- CS3263 - Foundations of Artificial Intelligence
- CS3264 - Foundations of Machine Learning
- CS5339 - Theory and Algorithms for Machine Learning
- CS5340 - Uncertainty Modelling in AI
- CS5446 - AI Planning and Decision Making
- CS5242 - Neural Networks and Deep Learning
- Project Modules (FYP/CP4106)
  - AI driven Modern Web Crawling
  - Numerical accuracy in Bayesian Optimization
  - Visualising Machine Learning Algorithms
  - (Or propose your own)

Buddy Attendance Taking

1. [@] and Bonus declaration is to be done here; You should show bonus to Eric.
2. Attempted tutorial should come with proof (sketches, workings etc...)
3. Random checks may be conducted.
4. Guest student should come and inform me.

Figure 3: Buddy Attendance: https://forms.gle/q5Secb3dHshmXNXd7

References I