CS2109s - Tutorial 10

Eric Han

Nov 15, 2023

Student Feedback on Teaching (SFT)

Your feedback is important to me; optional, but highly encouraged:

Figure 1: NUS Student Feedback on Teaching - https://blue.nus.edu.sg/blue/

Student Feedback on Teaching (SFT)

NUS Student Feedback https://blue.nus.edu.sg/blue/ 27/Oct - 24/Nov:

- 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

Annoucements

Important admin

- Graded Midterm has been published on Softmark.
  - $\mu = 61, \sigma = 16, \max = 92$
  - Deadline for queries regarding your grades on Softmark is Saturday, 18 November.
  - There are 2 Softmarks - https://softmark.nus.edu.sg
- EXP issue for T8, T9; I have feedback to Prof and he is resolving it.

PS7 Feedback

- Question 7: Task 2.5: Which classes did the model misclassify?
  - How to read the confusion Matrix
- Question 9: Task 3.1.2: What augmentations did you choose?
  - Augmentation is to help the training
  - Not about whether the augmentation is useful in the context

Final Exams

- Given dataset (private): Preprocessing, Feature Engineering
- Try/Train a bunch of models
  - Can use compute cluster
- Evaluate the model: Cross validation, Hyper parameter tuning
- Submit model, upload model
  - Score on test data $>$ Grade (80%), documentation (20%)
  - Upload the **training code** to produce the model
  - Must fit cosmology time to train the model
- Cannot discuss with
  - other students
  - me, the TA
- There is a mock final: platform/details TBA.

Bonus Questions

In case you want to go and review some of our bonus questions, Wenzhong from TG04 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

Question 1

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**
    $\mu_k \leftarrow \frac{1}{|\{x^{(i)}|c^{(i)}=k\}|} \sum_{x \in \{x^{(i)}|c^{(i)}=k\}} x$

Question 1a [G]

Prove that the algorithm. . .

a. always produces a partition with a lower loss (monotonically decreasing)
b. always converges [1].
c. [@] 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...
    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 of length more than 1 where the next is always lower.
    e. So must converge in finite number of iterations.

Question 1b [G]

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 and choose the clusters with the minimum loss.

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 1c

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 1d

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	Distance to first medoid	Distance 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

Notice that 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. Typically for ties, we want to choose a deterministic tie-break strategy to avoid infinite loops or excessive iterations. For example, do not swap clusters in the case of a tie/only update medoid if there is a decrease in loss.

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.

Question 2

Given this dataset:

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

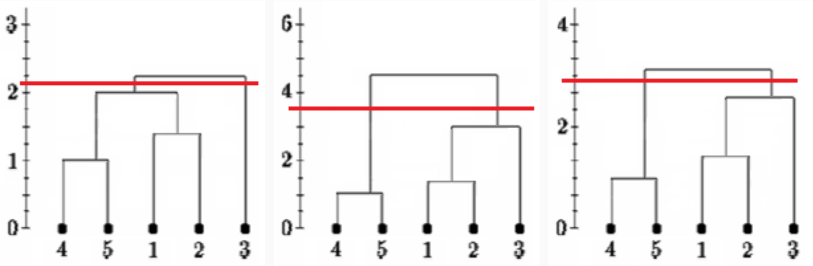
a. Complete the distance matrix.
b. Draw the dendrogram for the three linkage methods (Single, Complete and Centroid).

Recap

What is the algorithm to construct a hierarchical cluster?

Answer

	1	2	3	4	5
1	0				
2	$\sqrt{2}$	0			
3	$\sqrt{9}$	$\sqrt{5}$	0		
4	$\sqrt{10}$	$\sqrt{4}$	$\sqrt{13}$	0	
5	$\sqrt{17}$	$\sqrt{9}$	$\sqrt{20}$	$\sqrt{1}$	0



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

Question 3 [G]

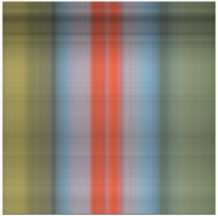
Using the `tut10.ipynb`, we study PCA:

a. The current choice of $k = 9$ does not produce a very nice output. What is a good value for k?
b. For the value of k you select in (a), what is the space saved by doing this compression?
c. What are the drawbacks of this form of compression?
d. [@] Can PCA be used for image recognition?
e. [@] How does JPEG work and relate to this technique?

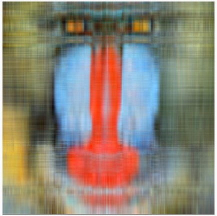
Recap

- What is PCA and how does it work?

Answer 3a



(a) 1/90.0/99.7



(b) 9/96.3/97.7

(c) 18/97.1/95.3

(d) 27/97.5/93.0

(e) 36/97.9/90.6

(f) 45/98.1/88.3

(g) 54/98.3/85.9

(h) 99/99.0/74.2

Figure 3: k/Variance/Space Saved

Answer 3b

When using $k = 99$, the (512×1536) 2D-array is now represented by U_{reduce} (512×99) and Z (99×1536). This demonstrates approximately 74% space saved.

$$\frac{(512 \times 99) + (99 \times 1536)}{512 \times 1536} = \frac{202752}{786432} = 0.258$$

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

Answer 3c

- 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.

Recommended Next Modules

- 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

Ask me anything

Ask me anything: https://www.menti.com/alpotusnviyg

Figure 4: Ask me anything

Buddy Attendance Taking

Take Attendance for your buddy: https://forms.gle/Ckkq639TNwWEx3NT6

1. Random checks will be conducted - `python ../checks.py TG0`

Figure 5: Buddy Attendance