Midterm Practice Solutions

Question 1

a) Calculate inverse of A.

$$\mathrm{A} = \begin{pmatrix} 3 & 5 \\ -7 & 2 \end{pmatrix}$$

det(A) = 6+35 = 41

$$A^{-1} = \frac{1}{det(A)} * \begin{pmatrix} 2 & -5 \\ 7 & 3 \end{pmatrix} = \begin{pmatrix} 2/41 & -5/41 \\ 7/41 & 3/41 \end{pmatrix}$$

b) Show A$A^{-1} = Identity$

$$\mathrm{A} = \begin{pmatrix} 3 & 5 \\ -7 & 2 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 2/41 & -5/41 \\ 7/41 & 3/41 \end{pmatrix}$$

$$AA^{-1} = \begin{pmatrix} 3 & 5 \\ -7 & 2 \end{pmatrix}\begin{pmatrix} 2/41 & -5/41 \\ 7/41 & 3/41 \end{pmatrix} = \begin{pmatrix} 6/41 + 35/41 & -15/41 + 15/41 \\ -14/41 + 14/41 & 35/41 + 6/41 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Hence Proved

Question 2

Calculate the cosine similarity between the following two sentences:

Sentence 1: “I like hamburger more than fried chicken.”
Sentence 2: “I like sushi more than hamburger.”

a)

WORDS	I	LIKE	HAMBURGER	MORE	THAN	FRIED	CHICKEN	SUSHI
VECTOR 1	1	1	1	1	1	1	1	0
VECTOR 2	1	1	1	1	1	0	0	1

b)

$$\mathrm{COS\ SIM\ (VECTOR\ 1, VECTOR\ 2)} = \frac{1 + 1 + 1 + 1 + 1 + 0 + 0 + 0}{\sqrt{7} * \sqrt{6}} = \frac{5}{\sqrt{42}}$$

Question 3

Given information

- P (Cloudy) = 0.6
- P (Rain) = 0.3
- P (Cloudy | Rain) = 0.5

Asked to Calculate

- $\mathrm{P\ (Rain\mid Not\ Cloudy)}\ = \frac{P\ (Rain{,}Not\ Cloudy)}{P\ (Not\ Cloudy)} = \frac{P\ (Not\ Cloudy\mid Rain)*P(Rain)}{P\ (Not\ Cloudy)}$

Needed probabilities

- P (Not Cloudy) = 1 - P (Cloudy) = 1 - 0.6 = 0.4
- P (Not Cloudy | Rain) = 1 - P (Cloudy | Rain) = 0.5

Result

- $\mathrm{P\ (Rain\mid Not\ Cloudy)}\ = \frac{0.5*0.3}{0.4} = 0.375$

<u>Since the probability of rain given it is not cloudy (0.375) is less than 0.45, it is still worth it to go to picnic.</u>

Question 4

Training classifiers involves estimating f: X -> Y, or P(Y|X)

a)

Generative models capture the joint probability P (X, Y) or just P(X) with no labels.

Example generative models (Naïve Bayes)

- Assume some functional form for P(X|Y), P(X)
- Estimate parameters of P(X|Y), P(X) directly from training data
- Use Bayes rule to calculate P (Y|X= xi)

b)

Discriminative models capture the conditional probability P (Y | X)

Example discriminative models (Logistic regression)

- Assume some functional form for P(Y|X)
- Estimate parameters of P(Y|X) directly from training data

c)

To classify different types of iris flower species from their attributes.

(i) <u>Using Generative Models:</u>

    - Estimate the parameters P (X | Y) from the training data

- Build a model of how data for a class looks like.
- Classify the data points based on learned P (X, Y) via bayes rule.
- Bayes classifiers use P (X, Y) to model P (Y | X) for classification. $P\ (y \mid x)\ = \frac{P\ (\ x \mid y)\ P\ (y)}{P\ (x)}$

(ii) <u>Using Discriminative Models:</u>

- Learn mappings directly from the space of inputs to iris flower species class labels {0, 1, 2}
- Try to learn P (Y | X) directly from training examples (Ex: Logistic Regression)

Question 5

Centroid/Partition based clustering (Ex: KMeans++)

- Select the points associated with the cluster
- Partition n objects into k clusters where each object belongs to exactly one cluster
- The number of clusters **k** is given in advance

Density based clustering (Ex: DBSCAN)

- Closely located points are put together
- Clustering based on density using local cluster criterion such as density connected points.
- This method can discover clusters of arbitrary shape and handle noise
- Can use DBSCAN algorithm to explain the clustering.

Distribution based clustering (Ex: Gaussian Mixture Models)

- Points from similar distribution are put together
- Cluster points by the distributional differences in the data generating function
- GMM is a probabilistic model that assumes that all data points are generated from mixture of a finite number of gaussian distributions with unknown parameters
- We model the mixture proportions and covariance structure in the data

Hierarchical based clustering

- Merge points bottom up or top down
- Given n points in a d-dimensional space, the goal of hierarchical clustering is to create a sequence of nested partitions, which can be conveniently visualized via a tree or hierarchy of clusters, also called the cluster dendrogram.
- Two main algorithmic approaches to hierarchical clusters: agglomerative (bottom up) and divisive (top down)

Question 6

The intuition behind this approach is that spreading out the k initial cluster centers is a good thing: the first cluster center is chosen uniformly at random from the data points that are being clustered, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing cluster center.

The k-means++ initialization algorithm is as follows:

1. Choose one center uniformly at random among the data points.
   Using random function to select among the indices 0 to n-1 would work.
2. For each data point x not chosen yet, compute D(x), the distance between x and the nearest center that has already been chosen.
   For each point, compute the distance to all the chosen cluster centers and choose the center with minimum of distances. Use the squared distance to compute the D(x)**2 for each point.
3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to D(x)**2. Providing weights to the random function would help.
4. Repeat Steps 2 and 3 until k centers have been chosen.
5. Now that the initial centers have been chosen, proceed using standard k-means clustering.

For KMeans Clustering:

1. Initialize cluster centers according to above.
2. Repeat 3 and 4 until convergence
3. Assign labels to each data point based on cluster centers
4. Assign centroids based on the datapoint labels.
5. Return the final cluster centers

Question 7

$$Entropy(X) = -\sum_i p_i * log(p_i)$$

a)

The dataset on the right with (5 circles and 7 stars) has more entropy than one on the left (7 circles and 1 star)

b)

For Dataset on Left:

p1 = 7/8,

p2 = 1/8

$$\mathrm{Entropy\ (X1)} = -\mathrm{p1} * \log(\mathrm{p1}) - \mathrm{p2} * \log(\mathrm{p2}) = -\frac{7}{8} log(\frac{7}{8}) - \frac{1}{8} log(\frac{1}{8})$$

For Dataset on Right:
p1 = 5/12,
p2 = 7/12

$$\mathrm{Entropy\ (X2)} = -\mathrm{p1} * \log(\mathrm{p1}) - \mathrm{p2} * \log(\mathrm{p2}) = -\frac{5}{12} log(\frac{5}{12}) - \frac{7}{12} log(\frac{7}{12})$$

Question 8

a)

```
def hamming(s1, s2):
    assert len(s1) == len(s2)

    dist = 0
    # Checking number of mismatch character positions
    for i, j in zip(s1, s2):
        if i!=j:
            dist += 1
    return dist


print(hamming('karolin', 'kathrin'))
print(hamming('karolin', 'kerstin'))
print(hamming('kathrin', 'kerstin'))
print(hamming('0000', '1111'))
print(hamming('2173896', '2233796'))
```

b)

Example 1:
Case when there is mismatch in length of strings.
Hamming ('01111', '1111') -> hamming distance is not defined.

Example 2:
Case when there is shift in character positions.
Hamming ('athrink', 'kerstin') -> hamming distance is 7.
In this example, we just replaced first and last characters of kathrin and see that the hamming distance jumps from 3 to 7.