Concordia University

Computer Science and Software Engineering Department

COMP 6321

Machine Learning

Assignment4

Student ID: 27162391

Question: 1 (a)

Answer: VC Dimension : 1

For one sample situation, there are two cases, both of which can be covered using one-sided interval $[a, \infty)$.

When there are two samples, the worst-case scenario happen when the minus sample is to the right of plus sample, which will not be covered by $[a, \infty)$. Thus its VC dimension is 1.

(b)

Answer: VC Dimension: 2

For one-sided intervals $(-\infty, a]$ or $[a, \infty)$, we already know their vc dimension is at least 1 from (a). For two samples situation, there are two cases.

For three samples, the worst-case scenario happen when the distribution from left to right is plus, minus, plus; or minus, plus, minus.

(c)

Answer: VC Dimension: 2

From question (a) and (b) we can easily get the VC dimension for finite unions of one-sided intervals is at least 2. (All cases with 2 samples can be covered in (a) and (b), using only one one-sided interval)

For three samples, the worst-case scenario happens when the distribution from left to right is minus, plus, minus.

(d)

Answer: VC Dimension: 4

We should note that all the worst case scenario happens when the samples are distributed in a way that all neighbors of positive samples are negative and vice versa. Thus, we need only to consider this worst case of different number of samples.

For sample number equals to 4, below are the two worst cases, and both would be covered with two intervals:

For sample number equals to 4, worst-case happen when it's ordered in plus, minus, plus, minus, plus:

(e)

Answer: VC Dimension: 2*k

It would fail when sample numbers get to 2*k+1, where the samples are distributed with plus, minus, plus,….minus, plus. Just as the question 1(d).

Question 2:

Question 2 (a)

$\Rightarrow$ For function $f(P_i) = \ln \frac{1}{P_i} = -\ln P_i$,

its second derivative is :

$$\frac{\partial^2 f(P_i)}{\partial (P_i^2)} = \frac{1}{P_i^2} \geq 0,$$

Thus, f is a convex function and according to Jensen's inequality, we have the inequality function :

$$f\left(\sum_{i=1}^{M} \lambda_i x_i\right) \leq \sum_{i=1}^{M} \lambda_i f(x_i), \text{ where } \lambda_i \geq 0 \text{ and}$$

$$\sum_{i=1}^{M} \lambda_i = 1$$

According to the question, we make $\lambda_i = P(x_i)$, $f(x_i) = -\log(x_i)$, which is convex function as proved above.

Thus, we can derive $x_i = Q(x) \,/\, P(x)$ in the question. To adapt to the inequality function, we can adjust the KL-divergence formula to :

$$KL(P \| Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

$$= \sum_{x} P(x) \log \frac{1}{\frac{Q(x)}{P(x)}}$$

using Jensen's inequality as proved above, we can have:

$$\sum P(x) \log \frac{1}{\frac{Q(x)}{P(x)}} \geq \log\left(\sum_x P(x) \frac{Q(x)}{P(x)}\right)$$

$$= \log\left(\sum_x Q(x)\right)$$

$$= 0$$

combine the two formula together, we can get that:

$$KL(P\|Q) \geq 0$$

Question 2 (b)

⇒ KL Divergence will be 0 when P is same distribution as Q, which is $P(x) = Q(x)$.

Thus, $P(x) / Q(x) = 1$.

Logarithm of it will be 0. finally we would get a 0 for its summation.

Question 2 (c).

The maximum will be reached when any x made $Q(x) = 0$.

That is to say for some x, Q is impossible event. In that case, we would get infinite as the maximum.

Question 2 (d)

This is wrong. It can be proved with an example:

Below is an arbitrary case, where P and Q have 4 possible options and their probability of x equals to the 4 options are shown in the chart.

x	0	1	2	3
P	0.25	0.25	0.25	0.25
Q	0.1	0.4	0.1	0.4

Using the formula,

We can get:

$$KL(P\,||\,Q) = (0.5 \log 2.5 + 0.5 \cdot \log 0.625) \,||\, 0.10$$

$$KL(Q\,||\,P) = (0.2 \log 0.4 + 0.8 \log 1.6) \,||\, 0.08$$

The values of the two are not the same, which violates the equality.

Question 2 (e)

According to the definition, we can derive the left part as:

$$KL\left(P(x,y) \,||\, Q(x,y)\right)$$

$$= \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{Q(x,y)} \quad \ldots (1)$$

Using the product rule of joint probability, we can have:

$$P(x,y) = P(y|x)\, P(x)$$
$$Q(x,y) = Q(y|x)\, Q(x)$$

$$\ldots (2)$$

Using (2) into (1), the formula can be derived into:

$$\sum_x \sum_y P(x,y) \log \frac{P(x,y)}{Q(x,y)}$$

$$= \sum_x \sum_y P(y|x) P(x) \log \frac{P(y|x)\, P(x)}{Q(y|x)\, Q(x)}$$

$$= \sum_x \sum_y P(y|x) P(x) \log \frac{P(y|x)}{Q(y|x} +$$

$$\sum_x \sum_y P(y|x) P(x) \log \frac{P(x)}{Q(x)}$$

$$\ldots (3)$$

$$= \sum_{x} P(x) \left(\sum_{y} P(y|x) \log \frac{P(y|x)}{Q(y|x)} \right) +$$

$$\sum_{x} P(x) \log \frac{P(x)}{Q(x)} \left(\sum_{y} P(y|x) \right)$$

For the last row, the first part is KL-divergence between two conditional probability distribution, $P(y|x)$ and $Q(y|x)$, which is:

$$\sum_{x} P(x) \left(\sum_{y} P(y|x) \log \frac{P(y|x)}{Q(y|x)} \right)$$

$$= KL\left(P(y|x) \,||\, Q(y|x)\right)$$

For the right half part, since

$$\sum_{y} P(y|x) = 1$$

Thus we can further get:

$$\sum_{x} P(x) \log \frac{P(x)}{Q(x)} \left(\sum_{y} P(y|x) \right)$$

$$= \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

$$= KL\left(P(x) \,||\, Q(x)\right)$$

So far, we've proved that the left part is equal to the summation of right part.

It's proved correct.

Question 2 (f)

According to KL-divergence definition, the left part can be re-written as:

$$\arg\min_{\theta} KL(\hat{P} \,\|\, P_\theta)$$

$$= \arg\min_{\theta} \sum_{x} \hat{P}(x) \log \frac{\hat{P}(x)}{P_\theta(x)}$$

$$= \arg\min_{\theta} \left[\sum_{x} \hat{P}(x) \log \hat{P}(x) - \sum_{x} \hat{P}(x) \cdot \log P_\theta(x) \right]$$

Since $\hat{P}$ stands for the empirical distribution, which is based on counts for each value of x in this data, it has fixed value / distributio over x, and is dependent from θ, thus we have,

$$\arg\min_{\theta} \left[\sum_{x} \hat{P}(x) \log \hat{P}(x) - \sum_{x} \hat{P}(x) \log P_\theta(x) \right]$$

$$= \arg\max_{\theta} \sum_{x} \hat{P}(x) \cdot \log P_\theta(x)$$

Further, $\hat{p}(x)$ can be seen as a para-meter here since it's dependent from θ.

Thus we have:

$$\arg\max_{\theta} \sum_{x} \hat{p}(x) \log P_{\theta}(x)$$

$$= \arg\max_{\theta} \sum_{x} \log P_{\theta}(x)$$

So far, we've proved that the left part equals to the right part.
The two arguments will ~~b~~ generate same optimization result over θ.

Question 3:

Please check the file ***Kmeans.m***. The three input arguments are: ***testData-*** data loaded from *hw4.dat*; ***K*** the initial number of clusters, and is 8 in this question; ***initialCenter***- the initial centroids given in the question.

The three output: ***newCenter***- the final centers of different clusters; ***nearestVec***- this is a vector of size 210012*1, each of it stores the nearest center for the corresponding pixel; ***distanceIter***- a vector that stores the sum of squared distance from each pixel to the nearest centroid of each iteration.

MATLAB CODE:

```
Data=load('hw4-image.txt');

K=8;

initialCenter = [255,255,255;255,0,0;128,0,0;0,255,0;0,128,0;0,0,255;0,0,128;0,0,0];

[newCenter, nearestVec] = Kmeans(testData, K , initialCenter);
```

- **How many clusters there are in the end?**

There are 6 clusters in the end, which can be checked in the chart below.

- **The final centroids of each cluster:**

The chart records all the centroids information.

R	G	B
241.2296	238.6252	233.8629
194.4116	136.3331	90.94365
136.2656	61.08973	10.10385
NaN	NaN	NaN
157.2917	97.59398	51.4333
NaN	NaN	NaN
78.92744	37.10829	13.0707
25.978	23.23575	23.60599

- **The number of pixels associated to each cluster:**
  MATLAB CODE:

```
noVec=zeros(8,1);
for i=1:K
    noVec(i)=sum(nearestVec==i);
end
```

The result is shown in the chart below:

index	# of pixels
1	4930
2	15190
3	52535
4	0

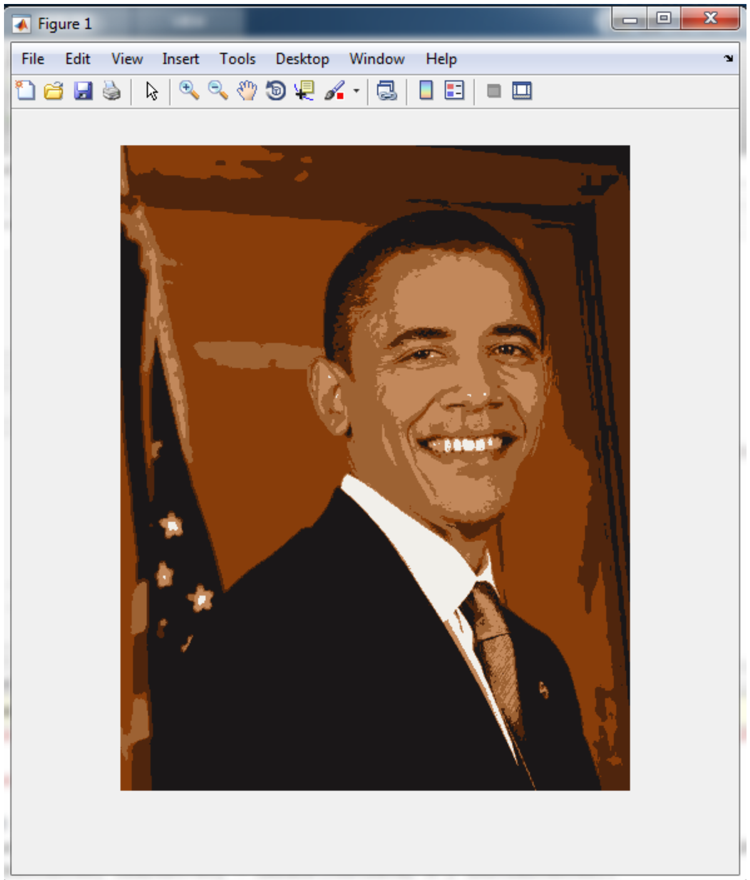
5	22075
6	0
7	40365
8	74917

- **The sum of squared distances from each pixel to the nearest centroid after every iteration of the algorithm:**

Below is a chart showing the sum of squared distances for all the pixels to their nearest centroid in total 50 iterations.

index	value	index	value	index	value	index	value	index	value
1	14442779	11	4393300	21	4390808	31	4135942	41	4135029
2	5799305	12	4394124	22	4387059	32	4135806	42	4135105
3	5347163	13	4395501	23	4378179	33	4135453	43	4135148
4	5187489	14	4395484	24	4356133	34	4135970	44	4135168
5	4765128	15	4395365	25	4292430	35	4135470	45	4135189
6	4423282	16	4395379	26	4222993	36	4135199	46	4135182
7	4385986	17	4395338	27	4184325	37	4134699	47	4135174
8	4386233	18	4394996	28	4159897	38	4134667	48	4135182
9	4388704	19	4394155	29	4144723	39	4134824	49	4135182
10	4391991	20	4393061	30	4137972	40	4134930	50	4135182

The visualized image is shown below:



Question 4:

		K-medoids	K-means
Similarity		The K-medoids algorithm shares the properties of K-means that each iteration decreases the error, will always converges, different initial center will give different local optimization, instead of global minimum.	
Difference	**Speed**	Converge slow	Converge fast, by comparison
	Noise	Good to lower variance	Bad for outliers
	Cluster number	Number of clusters will not change after calling this algorithm	Some cluster might disappear during clustering

The algorithm for K-medoids can be checked in file ***kMedoids.m***. Note that all the arguments mean the same as in method Kmeans.

References:

- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006
- Kevin P. Murphy, *Machine Learning. A Probabilistic Perspective,* MIT Press, 2012.
- https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence
- http://www.csse.monash.edu.au/~lloyd/tildeMML/KL/
- Pattern Recognition and Machine Learning, Bishop, Page 56
- A note from friend for the equations.