COMS 4771 FA18 HW4

Due: Fri Dec 07, 2018 at 11:59pm

You are allowed to write up solutions in groups of (at max) three students. These group members don't necessarily have to be the same from previous homeworks. Only one submission per group is required by the due date on Gradescope. Name and UNI of all group members must be clearly specified on the homework. No late homeworks are allowed. To receive credit, a typesetted copy of the homework pdf must be uploaded to Gradescope by the due date. You must show your work to receive full credit. Discussing possible solutions for homework questions is encouraged on piazza and with peers outside your group, but every group must write their own individual solutions. You must cite all external references you used (including the names of individuals you discussed the solutions with) to complete the homework.

1 **[PAC learning and VC dimension]**

(a) Let $\mathcal{F}$ be a set of binary classifiers induced by axis-parallel rectangles in D dimensions (where the interior is labelled negative and the exterior is labelled positive). State (and prove) the tightest possible bound on the VC-dimension of $\mathcal{F}$.

(b) Let $\mathcal{F}$ be from part (a). Define

- $\mathcal{G}_1 := \{g \mid g = \texttt{not}(f), \text{ where } f \in \mathcal{F}\}$,
- $\mathcal{G}_2 := \{h \mid h = h_1 \cup h_2, \text{ where } h_1, h_2 \in (\mathcal{F} \cup \mathcal{G}_1)\}$,
- $\mathcal{G}_3 := \{h \mid h = h_1 \cap h_2, \text{ where } h_1, h_2 \in (\mathcal{F} \cup \mathcal{G}_1)\}$.

State (and prove) the tightest possible bound on the VC-dimension of $\mathcal{F} \cup \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$.

(c) Consider a variant of the PAC model in which the learner is given *two* sets of examples: $S_+ = \{x_1^+, x_2^+, \ldots, x_m^+\}$ and $S_- = \{x_1^-, x_2^-, \ldots, x_n^-\}$, one is the set of positive examples and the other the set of negative examples, both distributed according to the underlying distribution $\mathcal{D}$ on $X \times \{-1, +1\}$. In other words, these two sample sets S_+ and S_- are drawn from distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ respectively, where

$$\mathcal{D}^+ \propto \mathcal{D}(x, 1) \text{ and } \mathcal{D}^- \propto \mathcal{D}(x, -1).$$

The learning algorithm is now given as input ϵ, δ, and the two sample sets, and it must return a hypothesis $h \in H$ such that with probability at least $1 - \delta$, both $\Pr_{x \in \mathcal{D}^+}[h(x) = -1] \leq \epsilon$ and $\Pr_{x \in \mathcal{D}^-}[h(x) = 1] \leq \epsilon$. Suppose H includes h^+, the always-plus concept, and h^-, the always-minus concept.

Show that H is efficiently PAC-learnable (in the standard one sample model) if and only if H is efficiently PAC-learnable in this two sample model.

2 **[Studying k-means]** Recall that in k-means clustering we attempt to find k cluster centers $c_j \in \mathbb{R}^d, j \in \{1, \dots, k\}$ such that the total (squared) distance between each datapoint and the nearest cluster center is minimized. In other words, we attempt to find $c_1, ..., c_k$ that minimizes

$$\sum_{i=1}^{n} \min_{j \in \{1,\dots,k\}} \|x_i - c_j\|^2, \tag{1}$$

where n is the total number of datapoints. To do so, we iterate between assigning x_i to the nearest cluster center and updating each cluster center c_j to the average of all points assigned to the jth cluster (aka Lloyd's method).

(a) **[it is unclear how to find the best k, i.e. estimate the correct number of clusters!]** Instead of holding the number of clusters k fixed, one can think of minimizing (1) over both k and c. Show that this is a bad idea. Specifically, what is the minimum possible value of (1)? what values of k and c result in this value?

(b) **[suboptimality of Lloyd's method]** For the case $d = 1$ (and $k \geq 2$), show that Lloyd's algorithm is *not* optimal. That is, there is a suboptimal setting of cluster assignment for some dataset (with $d = 1$) for which Lloyd's algorithm will not be able to improve the solution.

(c) **[improving k-means quality]** k-means with Euclidean distance metric assumes that each pair of clusters is linearly separable (see part ii below). This may not be the desired result. A classic example is where we have two clusters corresponding to data points on two concentric circles in the $\mathbb{R}^2$.

  (i) Implement Lloyd's method for k-means algorithm and show the resulting cluster assignment for the dataset depicted above. Give two more examples of datasets in $\mathbb{R}^2$, for which optimal k-means setting results in an undesirable clustering. Show the resulting cluster assignment for the two additional example datasets.

  (ii) Show that for $k = 2$, for any (distinct) placement of centers c_1 and c_2 in $\mathbb{R}^d$, the cluster boundary induced by minimizing the k-means objective (i.e. Eq. 1) is necessarily linear.

One way to get a more *flexible* clustering is to run k-means in a transformed space. The transformation and clustering is done as follows:

- Let G_r denote the r-nearest neighbor graph induced on the given dataset (say the dataset has n datapoints), that is, the datapoints are the vertices (notation: v_i is the vertex associated with datapoint x_i) and there is an edge between vertex v_i and v_j if the corresponding datapoint x_j is one of the r closest neighbors of datapoint x_i.
- Let W denote the $n \times n$ edge matrix, where

$$w_{ij} = \mathbf{1}[\exists \text{ edge between } v_i \text{ and } v_j \text{ in } G_r].$$

- Define $n \times n$ diagonal matrix D as $d_{ii} := \sum_j w_{ij}$, and finally define $L = D - W$.
- Compute the *bottom* k eigenvectors/values of L (that is, eigenvectors corresponding to the k smallest eigenvalues). Let V be the $n \times k$ matrix of of the bottom eigenvectors. One can view this matrix V as a k dimensional representation of the n datapoints.
- Run k-means on the datamatrix V and return the clustering induced.

We'll try to gain a better understanding of this transformation V (which is basically the lower order spectra of L).

(iii) Show that for any vector $f \in \mathbb{R}^n$,

$$f^{\mathsf{T}} L f = \frac{1}{2} \sum_{ij} w_{ij} (f_i - f_j)^2.$$

(iv) L is a symmetric positive semi-definite matrix.

(v) Let the graph G_r have k connected components $C_1, \ldots, C_k$. Show that the $n \times 1$ indicator vectors $\mathbb{1}_{C_k}, \ldots, \mathbb{1}_{C_k}$ are (unnormalized) eigenvectors of L with eigenvalue 0. (the ith component of an indicator vector takes value one iff the vertex v_i is in the connected component)

Part (v) gives us some indication on why the transformation V (low order spectra of L) is a reasonable representation. Basically: (i) vertices belonging to the same connected component/cluster (ie, datapoints connected with a "path", even if they are located far away or form odd shapes) will have the same value in the representation $V = [\mathbb{1}_{C_1}, \ldots, \mathbb{1}_{C_k}]$, and (ii) vertices belonging to different connected component/cluster will have distinct representation. Thus making it easier for a k-means type algorithm to recover the clusterings.

(vi) For each of the datasets in part (i) (there are total three datasets), run this flexible version of k-means in the transformed space. Show the resulting clustering assignment on all the datasets. Does it improve the clustering quality? How does the choice of r (in G_r) affects the result?

(You must submit your code for parts (i) and (vi) to Courseworks to receive full credit.)

3 **[From distances to embeddings]** Your friend from overseas is visiting you and asks you the geographical locations of popular US cities on a map. Not having access to a US map, you realize that you cannot provide your friend accurate information. You recall that you have access to the relative distances between nine popular US cities, given by the following distance matrix D:

Distances (D)	BOS	NYC	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NYC	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0

Being a machine learning student, you believe that it may be possible to infer the locations of these cities from the distance data. To find an embedding of these nine cities on a two dimensional map, you decide to solve it as an optimization problem as follows.

You associate a two-dimensional variable x_i as the unknown latitude and the longitude value for each of the nine cities (that is, x_1 is the lat/lon value for BOS, x_2 is the lat/lon value for NYC, etc.). You write down the an (unconstrained) optimization problem

$$\text{minimize}_{x_1,\ldots,x_9} \quad \sum_{i,j} \big(\|x_i - x_j\| - D_{ij}\big)^2,$$

where $\sum_{i,j}(\|x_i - x_j\| - D_{ij})^2$ denotes the embedding discrepancy function.

(i) What is the derivative of the discrepancy function with respect to a location x_i?

(ii) Write a program in your preferred language to find an optimal setting of locations $x_1, \ldots, x_9$. You must submit your code to Courseworks to receive full credit.

(iii) Plot the result of the optimization showing the estimated locations of the nine cities. (here is a sample code to plot the city locations in Matlab)

```
>> cities={'BOS','NYC','DC','MIA','CHI','SEA','SF','LA','DEN'};
>> locs = [x1;x2;x3;x4;x5;x6;x7;x8;x9];
>> figure; text(locs(:,1), locs(:,2), cities);
```

What can you say about your result of the estimated locations compared to the actual geographical locations of these cities?