COMS 4771 HW4 (Fall 2021)

Due: Dec 12, 2021 at 11:59pm

This homework is to be done **alone**. 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 the course discussion board and with your peers, but you must write their own individual solutions and **not** share your written work/code. You must cite all resources (including online material, books, articles, help taken from specific individuals, etc.) you used to complete your work.

1 Bayesian interpretation of ridge regression

Consider the following data generating process for linear regression problem in $\mathbb{R}^d$. Nature first selects d weight coefficients $w_1, \ldots, w_d$ as $w_i \sim N(0, \tau^2)$ i.i.d. Given n examples $x_1, \ldots, x_n \in \mathbb{R}^d$, nature generates the output variable y_i as

$$y_i = \sum_{j=1}^{d} w_j x_{i,j} + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$ i.i.d.

Show that finding the coefficients $w_1, \ldots, w_d$ that maximizes $P[w_1, \ldots, w_d | (x_1, y_1) \ldots, (x_n, y_n)]$ is equivalent to minimizing the ridge optimization criterion.

2 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

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

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 $x_1, \dots, x_9$. You must submit your code 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?

3 An alternate learning paradigm

In class you have seen that when building classifiers, one wants to minimize the expected classification error over a distribution $\mathcal{D}$. That is, we want to find the classifier f that minimizes:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\mathbf{1}[f(x) \neq y]\Big]. \qquad (1)$$

Since this quantity is not estimable in practice (since we don't know $\mathcal{D}$ and only have access to finite samples drawn from it), it is usually approximated via its empirical equivalent:

$$\frac{1}{|S|}\sum_{(x,y)\in S} \mathbf{1}[f(x) \neq y]. \qquad (2)$$

This latter quantity is the training error if S is the training set, and is the testing error if S is the testing set.

However for certain applications, obtaining a positively and negatively labelled samples S is not possible. Consider, for example, the problem of modelling user preferences based on news-feed that gets shown. Very simply, if a user interacts with a particular news item (such as they clicked and read it) shows that they are interested in the contents of the article, thus providing a positive label. But if a user does not interact with a particular news item, it is not clear whether the user dislikes the contents of the article, or simply didn't get around to viewing it. In such a scenario obtaining a good quality negatively labelled data sample is not possible. We thus need a slightly different learning paradigm where the training samples obtained are only either labelled as positive examples, or they are simply unlabeled examples. We can model this as follows:

- $\mathcal{D}$ is an unknown distribution over $\mathbb{R}^D \times \{0,1\} \times \{0,1\}$. $(x, y, s) \sim \mathcal{D}$ is a sample, where x is the input feature vector, y is the true label, and s (the "selection" variable) is whether x was interacted with (ie, selected) or not. Note that only x and s are observed.
- $\Pr[s = 1 \mid x, y = 0] = 0$, that is, a negatively labelled x is never selected.
- Given y, s and x are conditionally independent. That is, which x gets selected (given that, say, x positively labelled) is chosen independently.

The goal of this problem is to find an empirical estimator of (1) similar to (2) but using the unlabeled and positive data only.

1. Prove that $\Pr[y = 1 \mid x] = \frac{\Pr[s=1|x]}{\Pr[s=1|y=1]}$.
2. Using (i) prove that $\Pr[y = 1 \mid x, s = 0] = \frac{1-\Pr[s=1|y=1]}{\Pr[s=1|y=1]} \frac{\Pr[s=1|x]}{1-\Pr[s=1|x]}$.

   For the rest of the problem, assume that both quantities on the RHS can be estimated from (x, s) data only. This is trivially true for $\Pr[s = 1 \mid x]$ (since it does not depend on y). And while estimating $\Pr[s = 1 \mid y = 1]$ with only(x, s) data is nontrivial, it can be done under suitable conditions.
3. Letting p denote the PDF of $\mathcal{D}$ show that:

$$\begin{aligned}\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\mathbf{1}[f(x) \neq y]\big] = \int_x & p(x, s = 1)\mathbf{1}[f(x) \neq 1] \\ & + p(x, s = 0)(\Pr[y = 1 \mid s = 0, x]\mathbf{1}[f(x) \neq 1] \\ & \qquad + \Pr[y = 0 \mid s = 0, x]\mathbf{1}[f(x) \neq 0])dx\end{aligned}$$

4. Using parts (ii) and (iii) suggest an empirical estimator of (1) similar to (2) but that uses only (x, s) data.

   Hint: Try viewing unlabeled points as part positive and part negative. That is, replace unlabeled points by two "partial" points. One that is positive with weight $w(x)$ and one negative with weight $1 - w(x)$.

4 Exploring k-means in detail

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

$$\sum_{i=1}^{n} \min_{j \in \{1,\dots,k\}} \|x_i - c_j\|^2, \tag{3}$$

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 (3) over both k and c. Show that this is a bad idea. Specifically, what is the minimum possible value of (3)? what values of k and c result in this value?

(b) **[efficient optimal solution when either $k = 1$ or $d = 1$]** Optimizing the k-means objective (3) for the general case when $k \geq 2$ and $d \geq 2$ is NP-hard. But one can find an efficient optimial solution when either $k = 1$ or $d = 1$.

(i) What is the optimal objective value and the setting of the cluster centers in the case of $k = 1$?

(ii) 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.

(iii) **(note: this part is extra credit, only attempt it once everything else is done)** Propose an efficient algorithm (that is, an algorithm that runs in polynomial in n and k steps) for optimally solving k-means for $d = 1$ case.
(Hint: Use the fact that when $d = 1$, the dataset can be ordered as $x_{i_1} \leq x_{i_2} \leq \cdots \leq x_{i_n}$ and each cluster in the optimal assignment will be a nonoverlapping interval)

(c) **[kernelizing k-means]** k-means with Euclidean distance metric assumes that each pair of clusters is linearly separable. This may not be the case. 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.

Let ϕ denote an explicit non-linear feature space mapping in some inner product space. We will show how one can derive an *implicit* kernel-based version of the Lloyd's method for k-means, where we only operate on data as $\phi(x_i) \cdot \phi(x_j) = K(x_i, x_j)$.

(ii) Let z_{ij} be an indicator that is equal to 1 if the $\phi(x_i)$ is currently assigned to the jth cluster and 0 otherwise ($1 \leq i \leq n$ and $1 \leq j \leq k$). Show that the jth cluster center c_j can be written as $\sum_{i=1}^{n} \alpha_{ij}\phi(x_i)$, where α_{ij} only depends on z_{ij} variables.

(iii) Given any two data points $\phi(x_i)$ and $\phi(x_j)$, show that the square distance $\|\phi(x_i) - \phi(x_j)\|^2$ can be computed using only (linear combinations of) inner products.

(iv) Given the results of parts (ii) and (iii), show how to compute the square distance $\|\phi(x_i) - c_j\|^2$ using only (linear combinations of) inner products between the data points $x_1, \ldots, x_n$.

(v) From results from parts (ii), (iii) and (iv), propose the algorithm for kernelized version of Lloyd's method of finding k-means.

(vi) Implement your proposed kernelized k-means method and run it on the three example datasets of part (i). Compare the resulting cluster for various choices of kernels (e.g. linear kernel, quadratic kernel, rbf kernel).

(submit your datasets and kernelized code to receive full credit)