10-315 Intro to Machine Learning HW4

INSTRUCTIONS

- **Due: Wednesday, 25 November 2020 at 11:59 PM EDT.**
- **Format:** Complete this pdf with your work and answers. Whether you edit the latex source, use a pdf annotator, or hand write / scan, make sure that your answers (tex'ed, typed, or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.
- **How to submit:** Submit a pdf with your answers on Gradescope. Log in and click on our class 10-315, click on the appropriate *Written* assignment, and upload your pdf containing your answers. Don't forget to submit the associated *Programming* component on Gradescope if there is any programming required.
- **Policy:** See the course website for homework policies and Academic Integrity.

Name	
Andrew ID	
Hours to complete (both written and programming)?	

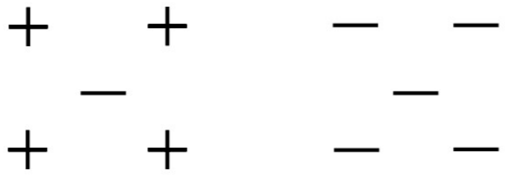
For staff use only

Q1	Q2	Q3	Q4	Total
/ 30	/ 24	/ 18	/ 28	/ 100

Q1. [30pts] Model Selection & Boosting

(a) [18pts] Model Selection

(i) [7pts] For the following dataset, calculate the Lean-One-Out Cross-Validation Error for 1-NN and 3-NN classifiers, and hence conclude which is the better model to use.



(ii) [3pts] In the above question, which model had higher model complexity, and why? Write 1 for 1-NN, and 3 for 3-NN in the smaller box. Use the bigger box for a brief, 1-2 sentence explanation.

Num:

Explanation:

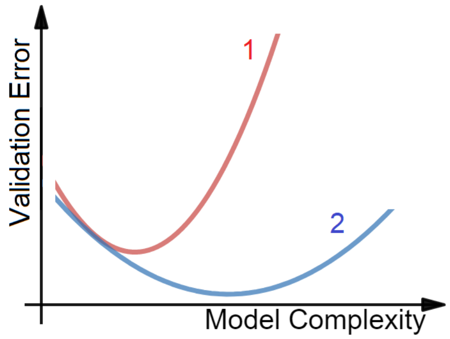
(iii) [3pts] Did the model with higher complexity achieve better results in the above problem? Explain why/why not, in 1-2 sentences.

Yes/No:

Explanation:

(iv) [5pts] The graph given below has 2 curves, each corresponding to a different sized training data-set, from the same distribution. Based on your answers to the previous questions, which curve do you expect corresponds to the larger data-set, and why?

Num:

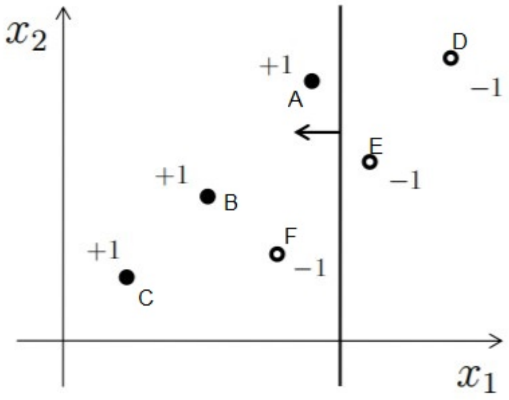


(b) [12pts] Boosting

Consider building an ensemble of decision stumps G_m with the AdaBoost algorithm,

$$f(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$$

The below figure displays a few labeled points in two dimensions as well as the first stump we have chosen. A decision stump predicts binary ±1 values, and depends only on one coordinate value (the split point). The little arrow in the figure is the normal to the stump decision boundary indicating the positive side where the stump predicts +1. All the points start with uniform weights.



(i) [2pts] List all the points whose weight will increase as a result of incorporating the first stump.

Answer:

(ii) [4pts] Give a possible stump that we could select at the next boosting iteration. Give its orientation (horizontal line/ vertical line), and describe what points are on either side of it. (For example, "Horizontal line; points above: A, D; points below: B, C, E, F")

Answer:

(iii) [6pts] Will the second stump receive higher coefficient in the ensemble than the first? In other words, will $\alpha_2 > \alpha_1$? Briefly explain your answer. (no calculation should be necessary).

Answer:

Q2. [24pts] Kernel PCA

(a) [4pts] PCA in high-dimensional feature spaces

Recall that for standard PCA, the principal components for a zero-centered dataset $\mathbf{X} \in \mathbb{R}^{N \times D}$ are found by eigen decomposition of its covariance matrix $\mathbf{C}$. (Note that N is the number of samples, and D is the number of features).

Let $\mathbf{x}_i \in \mathbb{R}^D$ be the i^{th} row of $\mathbf{X}$ as a column vector, and the covariance matrix can be expressed as:

$$\mathbf{C} = \frac{1}{N} \sum_{i}^{N} \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}}.$$

The j^{th} eigenvector (principal component) $\mathbf{u}_j$ and its corresponding eigenvalue λ_j can be found by solving,

$$\mathbf{C}\mathbf{u}_j = \lambda_j \mathbf{u}_j$$

Now let us consider a mapping $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$, which projects each original data point onto a higher dimensional space ($M > D$). This space is called **feature space**. It may be possible to obtain better dimensionality reduction when PCA is applied in the (nonlinear) feature space.

First assume that the new data points after projection are also zero-centered, meaning

$$\sum_{i}^{N} \phi(\mathbf{x}_i) = \mathbf{0}$$

and they have $\mathbf{S}$ as the covariance matrix

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^{\top}.$$

If we do standard PCA directly in feature space by solving the eigen decomposition problem,

$$\mathbf{S}\mathbf{v}_j = \lambda_j \mathbf{v}_j,$$

what could potentially be problematic? Explain with just one or two sentences.

Hint: think about the size of $\mathbf{S}$. Please provide a brief explanation.

In the subsequent problems, we will derive an alternate solution using kernels where the PCA in feature space can be achieved by eigendecomposition of an $N \times N$ matrix, instead of the $M \times M$ covariance matrix S.

(b) [6pts]

Show that the j^{th} principal component $\mathbf{v}_j$ can be expressed as a linear combination of transformed data points. That is, there exists an N-dimensional vector $\mathbf{w}_j = (w_{j1}, \ldots, w_{jn}, \ldots, w_{jN})^{\mathrm{T}}$ such that:

$$\mathbf{v}_j = \sum_{i=1}^{N} w_{ji} \boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\phi}(\mathbf{X})^{\top} \mathbf{w}_j$$

where $\boldsymbol{\phi}(\mathbf{X}) = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), ..., \boldsymbol{\phi}(\mathbf{x}_N)]^T$ is the $N \times M$ transformed data matrix.

Notice that this is akin to the trick we used for kernelizing logistic regression and linear/ridge regression. Now you will show that the weight vector $\mathbf{w}_j, \forall j$ can be found by solving a $N \times N$ eigen decomposition problem. The idea here is to replace the covariance matrix $\mathbf{S}$ with kernel matrix

$$\mathbf{K} \in \mathbb{R}^{N \times N}, \mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_j),$$

to avoid working in the feature space directly.

(c) [8pts]

Prove that any $\mathbf{w}_j$, of which the corresponding eigenvalue λ_j is non-zero, can be obtained by solving,

$$\mathbf{K}\mathbf{w}_j = N\lambda_j\mathbf{w}_j$$

Hint: start from the original eigen decomposition problem in terms of $\mathbf{S}$ and $\mathbf{v}_j$ and use the results from the previous questions to arrive at the above equation. Also, you may assume $\mathbf{K}$ is invertible.

(d) [6pts]

Notice that recovering the PC vector $\mathbf{v}_j$ from the weight vector $\mathbf{w}_j$ requires writing out the high-dimensional feature representation $\phi(\mathbf{X})$, which is problematic. In kernel PCA we avoid computing the PC vectors, and instead directly work with projections of data points onto the PC components. Show that the projection of any transformed point $\boldsymbol{\phi}(\mathbf{x})$ onto the PC $\mathbf{v}_j$ can be computed using the kernel and $\mathbf{w}_j$ only as:

$$\boldsymbol{\phi}(\mathbf{x})^\top \mathbf{v}_j = [k(\mathbf{x}, \mathbf{x}_1)\ k(\mathbf{x}, \mathbf{x}_2) \ldots k(\mathbf{x}, \mathbf{x}_N)]\mathbf{w}_j$$

Q3. [18pts] Programming: K-means

The following questions should be completed after you work through the programming portion of this assignment.

(a) [6pts] K=2

Include the images of the cluster centers after running k-means with two clusters.

Centers K=2:

(b) [6pts] K=5

Include the images of the cluster centers after running k-means with five clusters.

Centers K=5:

(c) [6pts] K=10

Include the images of the cluster centers after running k-means with ten clusters.

Centers K=10:

Q4. [28pts] Programming: PCA and GMM

The following questions should be completed after you work through the programming portion of this assignment.

(a) [12pts] PCA

Include the plots of the toy dataset before and after running PCA with K=2.

PCA before and after:

Include the plots of the MNIST zeros and ones dataset after running PCA with K=2.

PCA MNIST:

(b) [8pts] GMM Toy Datasets

Include the plots of after learning the GMM parameters for K=2 on toy dataset one and two.

GMM Toy 1, K=2:

GMM Toy 2, K=2:

(c) [8pts] GMM MNIST Zeros and Ones

Include the plots of after learning the GMM parameters for K=2 and K=5 on the MNIST zeros and ones dataset.

GMM MNIST, K=2:

GMM MNIST, K=5: