CS 412: Fall'21
Introduction To Data Mining

Take-Home Final

(Due Saturday, Dec 11, 6:00 pm)

General Instructions

- You will have to answer the questions yourself, you cannot consult with other students in class. It is an open book exam, so you can use the textbook and the material shared in class, e.g., slides, lectures, etc.
- The take-home Final will be due at 6 pm, Sat, December 11. We will be using gradescope for the Final. Contact the TAs if you are having technical difficulties in submitting the assignment. We will NOT accept late submissions.
- Your answers should be typeset and submitted as a pdf. You cannot submit a hand-written and scanned version of your Final. You can only do math equations, tables, and figures by hand, then scan or take a picture, and include that in your typeset answers.
- You DO NOT have to submit (python) code for any of the questions.
- For the questions, you will not get full credit if you only give out a final result. Please show the necessary details, calculation steps, and explanations as appropriate.
- If you have clarification questions, please use slack (preferred) or email all of us (instructor + 4 TAs). However, since the Final needs to be submitted within 24 hours, please ask the question as early as possible (ideally, within 1-2 hours after the Final is posted).

1. (25 points) Consider the following dataset for 2-class classification (Figure 1), where the blue points belong to one class and the orange points belong to another class. Each data point has two features $\mathbf{x} = (x_1, x_2)$. We will consider learning support vector machine (SVM) classifiers on the dataset.

Figure 1: 2-class classification dataset.

(a) (7 points) Recall that the hard margin linear SVM learns a linear predictor $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ by solving the following optimization:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 \ ,$$
$$\text{such that} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \ , \ i = 1, \ldots, n.$$

Can we train a hard margin linear SVM on the given dataset? Clearly explain your answer.

(b) (8 points) Recall that the soft margin linear SVM learns a linear predictor $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+b$ using slack variables $\xi_i, i = 1, \ldots, n$, by solving the following optimization:

$$\min_{\mathbf{w},b,\{\xi_i\}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \ ,$$
$$\text{such that} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \ , \ \xi_i \geq 0 \ , \ i = 1, \ldots, n.$$

Can we train such a soft margin linear SVM, i.e., with slack variables ξ_i, on the given dataset? If we can train such a classifier, is it possible that all slack variables will be zero, i.e., $\xi_i = 0, i = 1, \ldots, n$ for the dataset? Briefly justify your answers.

(c) (10 points) Professor Quadratic Kernel claims that mapping each feature vector $\mathbf{x}^i = (x_1^i, x_2^i)$ to a 6-dimensional space given by

$$\phi(\mathbf{x}^i) = [1 \quad x_1^i \quad x_2^i \quad x_1^i x_2^i \quad (x_1^i)^2 \quad (x_2^i)^2]^T$$

and training a linear hard-margin SVM in that mapped space would give a highly accurate predictor. Do you agree with Professor Kernel's claim? Clearly explain your answer.

2. (25 points) We consider comparing the performance of two classification algorithms A_1 and B_1 based on k-fold cross-validation. The comparison will be based on a t-test to assess statistical significance with significance level $\alpha = 5\%$. The null hypothesis is that the mean accuracy of the two algorithms A_1 and B_1 are exactly the same.

   (a) (5 points) We will assess the results for $k = 10$-fold cross-validation. What should be the degrees of freedom for the test? Briefly explain your answer.

   (b) (10 points) The accuracies for $k = 10$-fold cross-validation from two algorithms A_1 and B_1 are given in Table 1.

	1	2	3	4	5	6	7	8	9	10
A_1	0.908	0.962	0.878	0.956	0.939	0.955	0.944	0.933	0.881	0.949
B_1	0.449	0.585	0.381	0.433	0.475	0.430	0.520	0.590	0.565	0.443

Table 1: Accuracies on 10-folds for Algorithms A_1 and B_1.

   Is the performance of one of the two algorithms significantly different than the other based a t-test at significance level $\alpha = 5\%$? Clearly explain your answer by showing details of (a) the computation of the t-statistic, and (b) the computation of the p-value. Given the t-statistic `t_stat` and degrees of freedom `df`, you should be able to compute the p-value using the following:[1]

```
from scipy.stats import t
p_val = (1-t.cdf(abs(t_stat), df)) * 2
```

   (c) (10 points) Suppose we have a better version of algorithm B_1 called B_2. The accuracies for $k = 10$-fold cross-validation from algorithms A_1 and B_2 are given in Table 2.

	1	2	3	4	5	6	7	8	9	10
A_1	0.908	0.962	0.878	0.956	0.939	0.955	0.944	0.933	0.881	0.949
B_2	0.968	1.000	0.950	0.994	0.989	0.989	1.000	0.994	0.966	0.966

Table 2: Accuracies on 10-folds for Algorithms A_1 and B_2.

   Is one of the two algorithms significantly better than the other based a t-test at significance level $\alpha = 5\%$? Clearly explain your answer by showing details of (a) computation of the t-statistic, and (b) computation of the p-value.

[1] Alternatively, you can look a table for p-values for t-statistic, similar to how you had done it for the χ^2-statistic earlier in the semester.

3. (25 points) Let $\mathcal{Z} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^n, y^n)\}, \mathbf{x}^i \in \mathbb{R}^d, y^i \in \{0, 1\}, i = 1, \ldots, n$ be a set of n training samples. The input $\mathbf{x}^i, i = 1, \ldots, n$ are d-dimensional features, and x^i_j denotes the j-th feature of the i-th data point $\mathbf{x}^i$. The output $y^i \in \{0, 1\}, i = 1, \ldots, n$, are the class labels. We consider training a single layer perceptron where for any input $\mathbf{x}^i$, the output is given by

$$\hat{y}^i = g(a^i) = g(\mathbf{w}^T \mathbf{x}^i) = g\left(\sum_{j=1}^{d} w_j x^i_j\right) ,$$

where $g(a) = \max(a, 0)$, i.e., the ReLU activation function, and $a^i = \mathbf{w}^T \mathbf{x}^i$ is the input to the activation function. Note that the parameters $\mathbf{w} = [w_1 \cdots w_d]^T$ are the unknown parameters of the model. Consider a learning algorithm which focuses on minimizing squared loss between the true and predicted outputs:

$$L(\mathbf{w}|\mathcal{Z}) = \frac{1}{2}\sum_{i=1}^{n}(y^i - \hat{y}^i)^2 = \frac{1}{2}\sum_{i=1}^{n}(y^i - g(\mathbf{w}^T \mathbf{x}^i))^2 .$$

(a) (15 points) The stochastic gradient descent (SGD) algorithm updates the parameters based on a random chosen point $(\mathbf{x}^i, y^i)$ in each step. Show that the SGD update for parameter w_j with step size η is of the form

$$w_j^{\text{new}} = w_j + \eta g'(a^i)(y^i - \hat{y}^i)x^i_j , \tag{1}$$

where $a^i = \mathbf{w}^T \mathbf{x}^i$, and the gradient of the ReLU activation function is

$$g'(a^i) = \begin{cases} 1 , & \text{if } a^i \geq 0 , \\ 0 , & \text{otherwise} . \end{cases} \tag{2}$$

(b) (10 points) Instead of using the ReLU activation function $g(a^i) = \max(a^i, 0)$, we now consider using two other transfer functions:

i. linear activation, $g(a^i) = a^i$, and
ii. sigmoid activation, $g(a^i) = \frac{1}{1+\exp(-a^i)}$.

How will you modify (2) above to get the SGD algorithm corresponding to these activation functions? Do you also need to modify (1) above for these activation functions? Clearly explain your answer.

4. (25 points) This question considers partitioning based clustering methods. In particular, parts (a) and (b) consider the k-medoids algorithm, and (c) considers the k-means and k-medians algorithms.

   (a) (10 points) Clearly describe the Partitioning Around Medoids (PAM) k-medoids clustering algorithm using pseudocode and a brief description of each of the steps.

   (b) (5 points) What is the computational complexity of the Partitioning Around Medoids (PAM) k-medoids clustering algorithm? Briefly justify your answer.

   (c) (10 points) What is the motivation behind using the k-medians algorithm instead of the k-means algorithm in certain situations? Is the k-medians algorithm more computationally demanding than k-means? Briefly explain your answer.