Homework Assignment 5

COGS 118A: Introduction to Machine Learning I

Due: 11:59pm, Sunday, Nov. 11st, 2018 (Pacific Time).

Instructions: Answer the questions below, attach your code, and insert figures to create a PDF file; submit your file via Gradescope. You may look up the information on the Internet, but you must write the final homework solutions by yourself.

Late Policy: 5% of the total points will be deducted on the first day past due. Every 10% of the total points will be deducted for every extra day past due.

Grade: ____ out of 100 points

1 (10 points) Multiple Choices

1. Which of the following statements is **false** regarding structural risk minimization?

   (A) It is a method to perform model selection, i.e., choosing an optimal classifier to reduce the test errors.

   (B) The goal is to balance fitting the training data against the model complexity.

   (C) Different algorithms often have different model complexities.

   (D) We always need to compute the testing error for each model to perform structural risk minimization.

2. Which of the following statements is **false** regarding cross validation?

   (A) It is a method to perform model selection, i.e., choosing the optimal parameters for a classifier.

   (B) It works for both regression and classification models.

   (C) Cross validation can be used to perform structural risk minimization.

   (D) To perform k-fold cross validation, the greater the k is, the more opitmal the result will be.

   Ans: D D

2 (20 points) Linear Discriminant Analysis

Linear discriminant analysis has many applications, such as dimensionality reduction and feature extraction. In this problem, we consider a simple task. In data file `lda.npy`, there are two classes: class 0 and class 1. The data are expressed as matrices X_0 for class 0 and X_1 for class 1. Each $X_j = [\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, \ldots, \mathbf{x}_n^{(j)}]$. Note that in this problem we use **column vector** $\mathbf{x}_i^{(j)}$ for a single data point to simplify the calculation. Please fill the blanks in skeleton code `HW5.ipynb` to solve the following sub-problems:

(a) Compute the mean for each class, μ_0 and μ_1.

(b) Compute the covariance matrix for each class, Σ_0 and Σ_1.

The Fisher's linear discriminant analysis is defined to maximize criterion function:

$$S(\mathbf{w}) = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{within}}} = \frac{\left(\mathbf{w}^\top \mu_0 - \mathbf{w}^\top \mu_1\right)^2}{\mathbf{w}^\top (\Sigma_0 + \Sigma_1)\mathbf{w}}$$

An optimal solution $\mathbf{w}^*$ is:

$$\mathbf{w}^* = (\Sigma_0 + \Sigma_1)^{-1}(\mu_0 - \mu_1)$$

(c) Find the optimal $\widetilde{\mathbf{w}}^*$ with unit length.

Hint: The optimal $\mathbf{w}^*$ above is unnormalized. To normalize $\mathbf{w}^*$ to unit length in order to get $\widetilde{\mathbf{w}}^*$, you need to divide $\mathbf{w}^*$ by $||(\Sigma_0 + \Sigma_1)^{-1}(\mu_0 - \mu_1)||_2$, which is the L_2 norm of $\mathbf{w}^*$.

(d) Compute the projection on $\widetilde{\mathbf{w}}^*$ for each data point. Plot such projected data points with original data points in one figure.

Hint: Suppose we have a data point $\mathbf{x} = (x_1, x_2)^\top$, here, the data point $\mathbf{x}$ and $\widetilde{\mathbf{w}}^*$ are both column vectors. The projection on vector $\widetilde{\mathbf{w}}^*$ for $\mathbf{x}$ is simply the dot product:

$$\mathbf{x}_{\text{projected}} = \widetilde{\mathbf{w}}^* \left((\widetilde{\mathbf{w}}^*)^\top \mathbf{x}\right)$$

Ans: See attached Jupyter Notebook

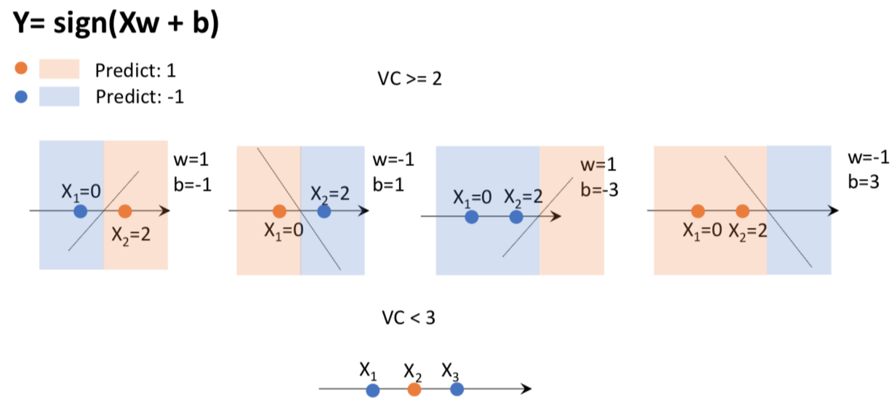
3 (10 points) Shattering

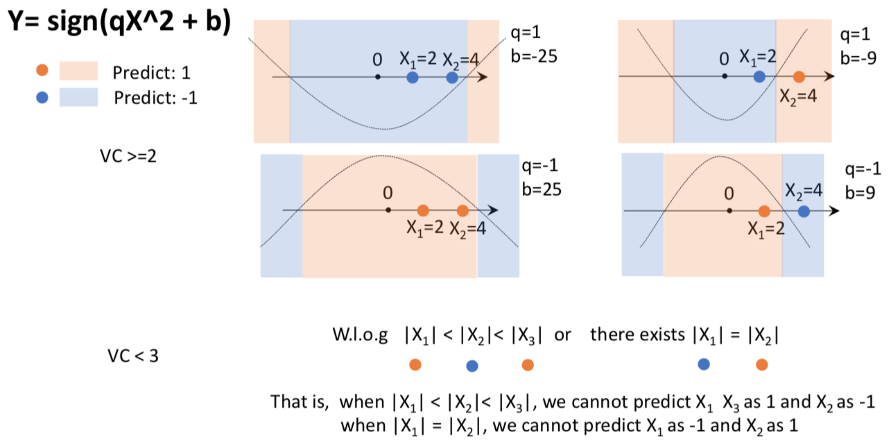
Use shattering to derive the VC-dimension for classifiers below. Show your work.

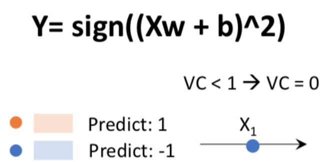
1) $f(x; w, b) = \text{sign}(x \times w + b)$

2) $f(x; w, b) = \text{sign}((x \times w + b)^2)$

where $x, w, q, b \in \mathbb{R}$, and w, q and b are free parameters.
Ans:





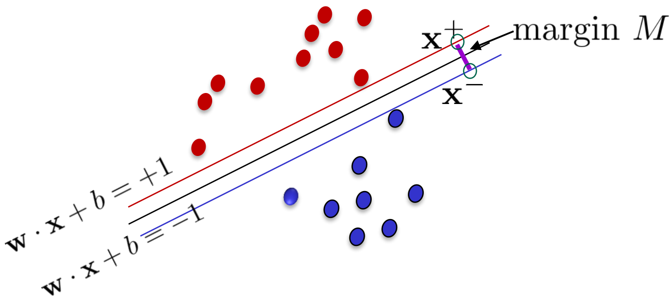


4 (10 points) Support Vector Machine 1

As shown in the figure, two boundaries are shifted to be parallel to the decision boundary in black, which is $\mathbf{w} \cdot \mathbf{x} + b = 0$. The equations of the boundaries are given in the figure. We first pick an arbitrary point $\mathbf{x}^-$ on the negative plane such that $\mathbf{w} \cdot \mathbf{x}^- + b = -1$; we then draw a line that passes $\mathbf{x}^-$ and is perpendicular to the negative plane; the intersection between this line and the positive plane can be denoted as $\mathbf{x}^+$ with $\mathbf{w} \cdot \mathbf{x}^+ + b = 1$. We thus have the following equations:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}^- + b &= -1, \\ \mathbf{w} \cdot \mathbf{x}^+ + b &= +1, \\ \mathbf{x}^+ &= \mathbf{x}^- + \lambda \mathbf{w}, \end{aligned}$$

where $\mathbf{x}^-$ is any point that lies on the blue boundary and $\mathbf{x}^+$ is any point that lies on the red boundary, $\mathbf{w}, b$ are given, and λ is an unknown parameter. Margin, M, is the distance between the two boundaries, which can be calculated as $M = ||\mathbf{x}^+ - \mathbf{x}^-||_2 = \sqrt{< \lambda\mathbf{w}, \lambda\mathbf{w} >}$. Please derive M to be parameterized by known parameters only (not containing λ).



Ans: (1) Derive λ based on the three equations given above
From given $\mathbf{w} \cdot \mathbf{x}^- + b = -1$,w $\cdot \mathbf{x}^+ + b = +1$,$\mathrm{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}^T$ we know that

$$\lambda \cdot \mathbf{w} = \mathbf{x}^+ - \mathbf{x}^-$$

If we multiple both side by w we would obtain

$$\mathbf{w} \cdot \lambda \cdot \mathbf{w}^T = \mathbf{w} \cdot \mathbf{x}^+ - \mathbf{w} \cdot \mathbf{x}^-$$

Together with

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}^- + b = -1 &\implies \mathbf{w} \cdot \mathbf{x}^- = -1 - b \\ \mathbf{w} \cdot \mathbf{x}^+ + b = +1 &\implies \mathbf{w} \cdot \mathbf{x}^+ = 1 - b \end{aligned}$$

The previous equality then becomes

$$\begin{aligned} \mathbf{w} \cdot \lambda \cdot \mathbf{w}^T &= \lambda \cdot \mathbf{w} \cdot \mathbf{w}^T = \lambda \cdot < \mathbf{w}, \mathbf{w} > \qquad\qquad (\lambda \text{ is a constant}) \\ &= \mathbf{w} \cdot \mathbf{x}^+ - \mathbf{w} \cdot \mathbf{x}^- \\ &= (1 - b) - (-1 - b) = 2 \end{aligned}$$

Solve it returns $\lambda = \frac{2}{<\mathbf{w}\cdot\mathbf{w}>}$
(2) Plug in the value of λ you derive in (1) to $M = ||\mathbf{x}^+ - \mathbf{x}^-||_2 = \sqrt{<\lambda\mathbf{w}, \lambda\mathbf{w}>}$.
Since $\lambda = \frac{2}{<\mathbf{w},\mathbf{w}>}$ and the L2 norm of w is $||\mathbf{w}||_2 = \sqrt{<\mathbf{w}, \mathbf{w}>}$

$$\begin{aligned}
M &= ||\mathbf{x}^+ - \mathbf{x}^-||_2 \\
&= \sqrt{<\lambda\mathbf{w}, \lambda\mathbf{w}>} \\
&= \sqrt{\lambda^2 <\mathbf{w}, \mathbf{w}>} \\
&= \sqrt{\frac{2*2}{<\mathbf{w}, \mathbf{w}><\mathbf{w}, \mathbf{w}>} <\mathbf{w}, \mathbf{w}>} \\
&= \frac{2\sqrt{<\mathbf{w}, \mathbf{w}>}}{<\mathbf{w}, \mathbf{w}>} = \frac{2}{\sqrt{<\mathbf{w}, \mathbf{w}>}}
\end{aligned}$$

5 (20 points) Support Vector Machine 2

In this problem, you are required to solve a series of questions using support vector machine (SVM). You will use Arrhythmia dataset that contains 452 data points. Each data point has a 279-dimensional feature vector and an 1-dimensional label (either 0 or 1), which means it is a binary classification task and can be solved by SVM. Please download the `arrhythmia.npy` as data source and `HW5.ipynb` to fill the blanks. You can use the functions from `sklearn` in your implementation unless in some case we ask you to implement a few built-in functions by yourself.

In this problem, you need to use the linear SVM to conduct the binary classification.

1) Load data from `arrhythmia.npy` and randomly shuffle the data points.

2) Select 80% of the data points as your **training and validation set**. The rest 20% is regarded as your **test set**. Actually, in the cross-validation, the training and validation set can be called as "training set". However, in order to be consistent with the code, we still call it "training and validation set" here.

3) Train the SVM classifier using a linear kernel. In linear SVM, there is a parameter C which adjusts the cost of outliers. You would need to use a grid search method to find the best parameter C^*. In fact, such grid search will utilize the cross-validation (3-fold) to get all the **average training accuracies** and **average validation accuracies** from the linear SVM model with different parameter C on training and validation set. The parameter $C = C^*$ which maximizes the **average validation accuracy** will be selected as the best. In fact, here "average" means the average accuracy over the folds in cross-validation, not the average accuracy over the different parameter C.

Hint 1: You are allowed to use `svm.SVC()` and `GridSearchCV()` in your code.

Hint 2: You can perform grid search on the following list of C:

$$C \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$$

4) Draw heatmaps for the result of grid search and find the best C^* for average validation accuracy. Report the heatmaps and best C^*.

5) Use the the best C^* to train a linear SVM classifier on training and validation set. Then, use the trained classifier to calculate the accuracy on test set. Report the test accuracy.

Ans: See attached Jupyter Notebook

6 (30 points) Implement Grid Search and Cross-validation

In this problem, you need to implement the grid search and cross-validation functions by yourself. You are **NOT** allowed to use `GridSearchCV()` here.

1) Implement a cross-validation function. In this function, you should divide your training and validation set into several subsets which have roughly the same size (the number of subsets is given by variable `fold`). Train the SVM with linear kernel for `fold` rounds and each round choose one different subset as validation set and all the other data points (all the other `fold - 1` subsets) as training set. Calculate the **training accuracy** and **validation accuracy** every round. Finally, return the **average training accuracy** and **average validation accuracy** over all rounds.

2) Implement a grid search function. In this function you need to traverse all of C. For each C, you should call your implemented cross-validation function above to get the average training accuracy and average validation accuracy. Finally, you need to return **average training accuracy matrix** and **average validation accuracy matrix** for all combinations of C.

3) Like what you have done in SVM, perform your implemented grid search with cross-validation (3-fold) to find the best combination of parameter C^* Draw heatmaps for result of grid search and get the best C^*. Report the heatmaps and the best C^*.

Hint: You can compare your heatmaps with the heatmaps from `GridSearchCV()` in the above sub-problem to confirm the correctness of your implementation. Both heatmaps should share similar behavior.

Ans: See attached Jupyter Notebook