PEKING UNIVERSITY, GUANGHUA SCHOOL OF MANAGEMENT
Machine Learning and Artificial Intelligence

Assignment 2

Prof. Yingjie Zhang

Due date: Nov 7, 2021 11:59pm

Instruction

- This homework includes both conceptual questions and implementation questions (both are described in this .pdf file)
- Deliverables: Please submit **one pdf file** (with all your answers to the conceptual questions) and **python files** (including one with a .ipynb extension and one with a .html extension) that include all coding work (with necessary comments).
- You may discuss the questions with fellow students, however you must always write your own solutions and must acknowledge whom you discussed the question with. Do not copy from other sources, share your work with others, or search for solutions on the web. Plagiarism will be penalized according to university rules.

1 ADABOOST

We have a small dataset with eight data points. Each data point is represented with two features and one label $y \in \{+, -\}$:

$$\begin{aligned} &X^{(1)} = (-1, 0, -), X^{(2)} = (-0.5, 0.5, -), X^{(3)} = (0, 1, +), X^{(4)} = (0.5, 1, +), \\ &X^{(5)} = (1, 0, -), X^{(6)} = (1, -1, -), X^{(7)} = (0, -1, +), X^{(8)} = (0, 0, +) \end{aligned} \tag{1.1}$$

(a) Suppose we run an AdaBoost algorithm with three iterations. And the weak learner we consider in the AdaBoost is decision stumps (i.e., one-level decision tree: a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves)). For each iteration, $t = 1, 2, 3$, compute $\varepsilon_t, \alpha_t, Z_t, D_t(i) \forall i,$, and draw the decision stump (hint: you can plot the eight points in one graph, and plot the linear separator for each iteration, as what we did in class).

(b) Compute the training error of the above AdaBoost. Briefly explain why AdaBoost outperforms a single decision stump.

2 K-MEANS CLUSTERING

Recall in class, we are choosing K cluster centres $c_j, j \in \{1, 2, ..., K\}$ to minimize:

$$\sum_{i=1}^{N} min_j \|x^{(i)} - c_j\|_2^2. \tag{2.1}$$

(a) Instead of treating K as a model hyperparameter (i.e., fix K), can we minimize 2.1 over both K and c simultaneously?

(b) Recall that in a kernelized algorithm, the solution can be expressed as a linear combination of training samples and the algorithm only replies on inner products between data points rather than their explicit representations. In K-Means Clustering, if we apply Euclidean distance metric, each pair of clusters is linearly separable. And we can still apply kernels to obtain a non-linear version.

1. Let z_{ij} be an indicator that is equal to 1 if the $x^{(i)}$ is currently assigned to the j^{th} cluster and 0 otherwise. Show that the j^{th} cluster center c_j can be updated as a format like $\sum_{i=1}^{n} \alpha_{ij} x^{(i)}$. Explicitly show how α_{ij} can be computed given all z's.

2. Given two data points $x^{(1)}$ and $x^{(2)}$, show that the square distance $\|x^{(1)} - x^{(2)}\|^2$ can be computed using only linear combinations of inner products.

3. Combining results from the previous two questions, show how to compute the square distance $\|x^{(i)} - c_j\|^2$ using only linear combinations of inner products between the data points $x^{(1)}, x^{(2)}, ..., x^{(N)}$.

3 SUPPORT VECTOR MACHINE

In Figure 3.1, there are different SVMs with different shapes/patterns of decision boundaries. The training data is labeled as $y_i \in \{-1, 1\}$, represented as the shape of circles and squares respectively. Support vectors are drawn in solid circles. Match the scenarios described below to one of the 6 plots (note that one of the plots does not match to anything). Each scenario should be matched to a unique plot. Explain in less than two sentences why it is the case for each scenario.

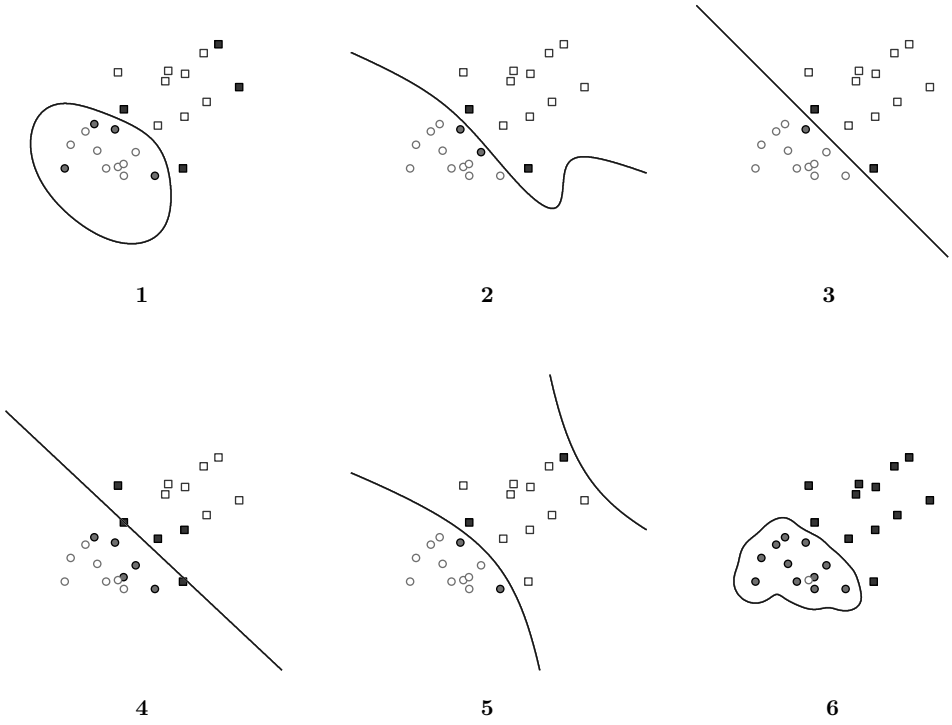

Figure 3.1: SVM boundaries

1. A soft-margin linear SVM with $C = 0.02$.
2. A soft-margin linear SVM with $C = 20$.
3. A hard-margin kernel SVM with $k(\mathbf{x}, \mathbf{z}) = x \cdot z + (\mathbf{x} \cdot \mathbf{z})^2$
4. A hard-margin kernel SVM with $k(\mathbf{x}, \mathbf{z}) = exp(-5||\mathbf{x} - \mathbf{z}||^2)$
5. A hard-margin kernel SVM with $k(\mathbf{x}, \mathbf{z}) = exp(-\frac{1}{5}||\mathbf{x} - \mathbf{z}||^2)$

4 IMPLEMENTATION: NAIVE BAYES

You will implement a Naive Bayes classifier for a text classification problem. The dataset is a collection of text articles, each coming from either the serious European magazine *The Economist*, or from the not-so-serious American magazine *The Onion*. The goal is to learn a classifier that can distinguish between articles from each magazine.

Download the data `'naive_bayes_data.zip'`. We have preprocessed the articles so that they are easier to use in your experiments. We extracted the set of all words that occur in any of the articles, known as the vocabulary. There are five files in the zip:

- `Vocabulary` is a V dimensional list that contains every word appearing in the documents. When we refer to the j^{th} word, we mean `Vocabulary(j)`.
- `XTrain` is a $n \times V$ dimensional matrix describing the n documents used for training your Naive Bayes classifier. The cell `XTrain(i,j)` is 1 if word j appears in the i^{th} training document and 0 otherwise.
- `yTrain` is a $n \times 1$ dimensional matrix containing the class labels for the training documents. `yTrain(i,1)` is 0 if the i^{th} document belongs to The Economist and 1 if it belongs to The Onion.
- Finally, `XTest` and `yTest` are the same as `XTrain` and `yTrain`, except instead of having n rows, they have m rows. This is the data you will test your classifier on and it should not be used for training.

Using Python to answer the following questions with explicit answers either shown in your ipython file or in a pdf report.

1. Train a Naive Bayes classifier. Produce a confusion matrix for the Naive Bayes classifier. Plot the matrix as a heatmap.
2. Calculate and report the precision and recall considering the articles from *The Onion* as the positive class.
3. Calculate and report the precision and recall considering the articles from *The Economist* as the positive class.
4. What is the misclassification rate of Naive Bayes on this problem?
5. What is the true class of the 38^{th} observation? What is its predicted class? What are the estimated posterior probabilities for the 38^{th} observation according to Naive Bayes?