Washington State University
School of Electrical Engineering and Computer Science
CptS 315 – Introduction to Data Mining
Online

Ananth Jillepalli

Homework 3

Name: Nam Jun Lee

Student Number: 11606459

Q1. Answer the following with a yes or no along with proper justification.

a. Is the decision boundary of voted perceptron linear?

No, the decision boundary of the voting perceptron is nonlinear. Because the voting perceptron can have multiple weighting vectors in the quadrant, it can cause a boundary, so the decision boundary can be nonlinear.

b. Is the decision boundary of averaged perceptron linear?

Yes, the decision boundary of the average perceptron is linear. The average perceptron has linear decision boundaries because all weights are averaged into one vector to spread the pointers more evenly.

Q2. Consider the following setting. You are provided with n training examples: $(x_1,y_1,h_1),(x_2,y_2,h_2),\cdots,(x_n,y_n,h_n)$, where x_i is the input example, y_i is the class label (+1 or -1), and $h_i > 0$ is the importance weight of the example. The teacher gave you some additional information by specifying the importance of each training example. How will you modify the perceptron algorithm to be able to leverage this extra information?

Additional information was obtained from the Panopto Hw3 video to focus on the learning rate. The Perceptron algorithm utilizing this additional information needs to track the weight of the survival vector, so it needs to add an updated weight to the Perceptron algorithm.

Q3. Consider the following setting. You are provided with n training examples: (x_1, y_1), (x_2, y_2), $\cdot\cdot\cdot$, (x_n, y_n), where x_i is the input example, and y_i is the class label (+1 or -1). However, the training data is highly imbalanced (say 90% of the examples are negative and 10% of the examples are positive) and we care more about the accuracy of positive examples. How will you modify the perceptron algorithm to solve this learning problem?

This training data has 90 percent negative and 10 percent positive. This means that the data is unbalanced. Model learning using these unbalanced datasets can never produce positive accuracy. Such disproportionate training data can cause overfitting, so data must be balanced either by oversampling the dataset or by undersampling. Another method is to set MaxIter, the only hyperparameter of the Perceptron algorithm.

Q4. You were just hired by MetaMind. MetaMind is expanding rapidly, and you decide to use your machine learning skills to assist them in their attempts to hire the best. To do so, you have the following available to you for each candidate i in the pool of candidates I: (i) Their GPA, (ii) Whether they took Data Mining course and achieved an A, (iii) Whether they took Algorithms course and achieved an A, (iv) Whether they have a job offer from Google, (v) Whether they have a job offer from Facebook, (vi) The number of misspelled words on their resume. You decide to represent each candidate $i \in I$ by a corresponding 6-dimensional feature vector $f(x^{(i)})$. You believe that if you just knew the right weight vector $w \in R^6$ you could reliably predict the quality of a candidate i by computing $w \cdot f(x^{(i)})$. To determine w your boss lets you sample pairs of candidates from the pool. For a pair of candidates (k, l) you can have them face off in a "DataMining-fight." The result is score ($k \succ l$), which tells you that candidate k is at least score ($k \succ l$) better than candidate l. Note that the score will be negative when l is a better candidate than k. Assume you collected scores for a set of pairs of candidates P. Describe how you could use a perceptron based algorithm to learn the weight vector w. Make sure to describe the basic intuition; how the weight updates will be done; and pseudo-code for the entire algorithm.

To learn the weight vector w, we initialize it as a random vector in a perceptron-based algorithm. If a particular vector belongs to the first item and the weight of the particular vector is less than zero, update w. Conversely, if this particular vector belongs to the second item and the weight of the particular vector is greater than 0, we subtract w. If neither, leave it as it is.

Pseudo-code:

```
w = set initialize random vector

L1 = 1 # item 1

L2 = 0 # item 2

For each feature vector:

    If feature vector in L1 and $w \cdot f(x^{(i)}) < 0$:

        w += feature vector

    Elif feature vector in L2 and $w \cdot f(x^{(i)}) >= 0$:

        w -= feature vector

    Else:

        PASS
```

Q5. Suppose we have n+ positive training examples and n– negative training examples. Let C+ be the center of the positive examples and C– be the center of the negative examples, i.e., $\boldsymbol{C_+} = \frac{\mathbf{1}}{\boldsymbol{n_+}} \sum_{i:yi=+1} \boldsymbol{x_i}$ and $\boldsymbol{C_-} = \frac{\mathbf{1}}{\boldsymbol{n_-}} \sum_{i:yi=-1} \boldsymbol{x_i}$. Consider a simple classifier called CLOSE that classifies a test example x by assigning it to the class whose center is closest.

1. **Show that the decision boundary of the CLOSE classifier is a linear hyperplane of the form sign(w · x + b). Compute the values of w and b in terms of C+ and C–.**

$C_+ = || \text{x} - C_+ ||^2$

$= || \text{x} ||^2 – 2\text{x}C_+ + || C_+ ||^2$

$C_- = || \text{x} – C_- ||^2$

$= || \text{x} ||^2 – 2\text{x}C_- + || C_- ||^2$

So, $|| \text{x} ||^2 – 2\text{x}C_+ + || C_+ ||^2 \leq || \text{x} ||^2 – 2\text{x}C_- + || C_- ||^2$

$= 2\text{x}C_+ - 2\text{x}C_- + ||C_+||^2 – ||C_-||^2 \geq 0$

Hence,

Value of $\text{w} = 2\ (C_+ - C_-)\text{x}$

Value of $\text{b} = ||C_+||^2 - ||C_-||^2$

2. **Recall that the weight vector can be written as a linear combination of all the training examples: $\boldsymbol{w} = \sum_{i=1}^{n_+ + n_-} \boldsymbol{\alpha_i} \times \boldsymbol{y_i} \times \boldsymbol{x_i}$. Compute the dual weights (α's). How many of the training examples are support vectors?**

$n_+ = 2 \sum_{i=1}^{n} xi$; $n_- = \sum_{i=1}^{-n} xi$

⇨ $\text{w} = 2 \sum_{i=1}^{n} xi - 2 \sum_{i=1}^{-n} xi$

Hence, dual weights $\alpha = \frac{2}{n_+}\ or\ \frac{2}{n_-}$

Therefore, 2 training examples are support vectors.

Q6. Please read the following paper and write a brief summary of the main points in at most TWO pages.

This article talks about the definition and problems of machine learning. Machine learning is one of the emerging technologies as the current era becomes digital. Machine learning is a system that focuses on classification and outputs a class that is a single discrete value by entering a vector of discrete or continuous values. Although the classifier's performance in well-organized data is excellent, there are three factors that can improve the classifier's performance in millions of data. The first is an optimization technique that is critical to the learner's efficiency, and it helps a lot in determining the classifier if there is an optimality between the two directors. The second is necessary to distinguish between good and bad classifiers by evaluation techniques. The final technique is to express the classifier as an expression in some official languages that the computer can process.

The basic goal of machine learning is generalization. The person who will create the classifier must learn the final classifier for the entire data, leaving some data aside and testing the selected classifier at the end. Failure to do so may result in classifier contamination by test data, which in turn may render the entire data itself useless.

Overfitting errors can often be encountered during machine learning. This takes an unambiguous form and is caused by noise in the training set. To avoid overfitting, it is necessary to select the optimal size of the cross-validation or decision tree to prevent overfitting or to add regularization terms to the evaluation function. However, this may lead to underfitting. The only way to prevent both at the same time is to learn the perfect classifier.

After proceeding with the overfitting step, the next problem is the curse of dimensionality. This is one of the problems that makes machine learning difficult, and the similarity-based reasoning on which machine learning algorithms depend is decomposed in higher dimensions. It is easy for classifiers to make classifiers from two to three dimensions, but it is very difficult to interpret the results from these high dimensions. Therefore, ordinary learners implicitly utilize low effective dimensions or use algorithms to explicitly reduce dimensionality

Machine learning projects do not succeed every time. Many independent functions that are correlated between each class are easy to learn, but on the contrary, if they are surrounded by very complex functions or have little correlation, it can lead to failure of machine learning. As a result, machine learning learners need technical skills, but they also need to know skills such as intuition, creativity, and black art. The best possible set of features should be configured to create a classifier. However, if we construct the best possible set of features and the classifier shows inaccurate results, we need to design better learning algorithms or collect more data. However, collecting and using a lot of data creates another problem. It is a time and memory. Although vast amounts of data are available, it takes too long to process data and is limited by memory overload.

In conclusion, machine learning projects are an important component of learners' design, and learners' expertise is important. Learners continue to need additional effort and an accurate analysis of classifiers obtained through their efforts will yield greater insights in the future.