CSE 353 – Homework III

Instructor: Dr. Ritwik Banerjee

Submission Deadline: May 12, 2017 (Sunday), 11:59 pm

1 Theory

I: Parameterized Cluster Distance **10 points**

Consider a dataset clustered into K clusters, where the i^{th} cluster C_i consists of n_i data points $(1 \leq i \leq K)$. Further, assume that there is some notion of a distance measure between two clusters C_i and C_j, denoted by $d_{i,j}$. Generally, if C_i and C_j are merged to form a new cluster $C_{i \cup j}$, then its distance to some other cluster C_k may not have a simple relation to $d_{i,k}$ and $d_{k,j}$. However, consider the equation, expressing it as a parameterized linear combination of the pre-merge weights:

$$d_{k,i\cup j} = \alpha_i d_{i,k} + \alpha_j d_{k,j} + \beta d_{i,j} + \gamma \left| d_{i,k} - d_{k,j} \right|$$

Show that if $\beta = 0$, then there exist real values of α_i, α_j, and γ such that[1]

$$d_{k,i\cup j} = \min(d_{i,k}, d_{k,j}) \text{ , and}$$
$$d_{k,i\cup j} = \max(d_{i,k}, d_{k,j})$$

II: Error Probability for k-NN **10 points**

Consider the special case where $k = 1$. Further, assume that we have two classes C_1 and C_2 with equal priors, *i.e.*, $P(C_1) = P(C_2) = 0.5$. Finally, you are given that the data is obtained from the two distributions

$$P(x|C_1) = \begin{cases} 2x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \qquad P(x|C_2) = \begin{cases} 2(1-x) & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

(a) What is the discriminant function (*i.e.*, the function that decides whether a test data point belongs to C_1 or C_2) for this classification?

(b) What is the classification error? [Hint: You need to find the total probability of errors.]

III: Simpler Computation in k-NN **10 points**

Computing distances in high dimensions may sometimes be prohibively expensive. An often-used technique is to do some distance-related computations in lower dimensions as preliminary filtering.

[1] Of course, both will never be simultaneously true, so you have to find different parameter values for each.

(a) Given $\mathbf{x} = \{x_1, \ldots, x_d\}$ and $\mathbf{y} = \{y_1, \ldots, y_d\}$, two vectors in a d-dimensional space, use the following inequality called **Jensen's inequality**

$$f(\mathbb{E}(z) \leq \mathbb{E}(f(z)) \text{ for all convex functions } f$$

to show that

$$\left[\frac{1}{\sqrt{d}}\sum_{i=1}^{d} x_i - \frac{1}{\sqrt{d}}\sum_{i=1}^{d} y_i\right]^2 \leq \sum_{i=1}^{d}(x_i - y_i)^2$$

(b) Explain what the above inequality means in terms of distance computations, and discuss how this property can be used to reduce the computational complexity of the process of finding the nearest neighbor of a test data point. This discussion need not be a formal proof.

IV: Linear Separability **5 points**

Suppose you have a dataset where the decision boundary is the d-dimensional *hyperellipse* given by

$$\sum_{i=1}^{d} \frac{x_i^2}{a_i^2} = 1$$

Find the transformation that maps the input space into the feature space such that the positive and negative examples become linearly separable in the feature space.

2 Programming & Experiments

In this third and final assignment of the semester, you are not required to write your own code from scratch for any algorithm. Instead, you should use two machine learning libraries to use support vector machines (SVM). The goal is to explore SVM for a specific learning task, and learn a model that does not overfit. Your model will be tested on a small dataset that will not be provided to you (we are effectively splitting a part of the original dataset, and keeping it for separate testing).

2.1 Dataset

The training data sample is available for download (link provided on our course web page). It is a single `.csv` file, 22.4 MB. **Please right-click and download. Otherwise your browser may crash while trying to open the file in preview**.

2.2 The learning task

I have never played DOTA 2, but I know people who know people who have. It is a game between two teams of 5 players each. Each team chooses 10 "heroes" out of a set of 113. The data provided is a single `.csv` file where each line corresponds to a single game, and consists of

Index 0: Team won the game (1 or -1)
Index 1: Cluster ID (related to location)
Index 2: Game mode
Index 3: Game type (e.g. "ranked")

Index 4 - end: Each element is an indicator for a hero. A value of 1 indicates that a player from team '1' played as that hero, and a value of -1 indicated that a player from the other team (i.e., the '-1' team) played as that hero (a hero can be selected by only one player each game). So, each row has five '1' and five '-1' values. All other values are 0, indicating that no player from either team chose that specific hero[2].

Your task is to build a model that can distinguish between the winning and losing teams, as specified by index-0. This consists of the following:

V: LibSVM 30 points

Read up on how to install and run LibSVM here: https://www.csie.ntu.edu.tw/~cjlin/libsvm/. Convert the given dataset into LibSVM's format, and perform 5-fold cross validation to observe the average accuracy across the folds. Your results are going to statistically insignificant if the accuracy is fluctuating more than 10%, so try to play around with the various parameters and repeat the experiments until you observe reasonably steady results. LibSVM allows you to save the model as a file, so once you obtain the best model, you should retain that file. This model file must be submitted along with your code.

For this first set of experiments, you may NOT use any other library. LibSVM is very well known, and there are certainly other libraries that provide functions to convert a dataset into LibSVM's format. This conversion must be your own code.

VI: k-means clustering 25 points

The second set of experiments requires you to either use Weka 3[3] (if you want to use Java) or `scikit-learn`[4] (if you want to use Python). This time, you have to use k-means clustering to create two clusters. That is, perform unsupervised learning instead of a supervised classification. You will have to write some wrapper code to use these libraries, but there is no need to implement the clustering algorithm on your own. Once the clustering is done, estimate its performance by computing the following:

(a) Percentage split of team '1' wins across two clusters (e.g., $x\%$ in cluster-1 and $y\%$ in cluster-2 (and similarly, also for team '-1').

VII: Final Report 10 points

Your final report must contain the following details in some easy-to-understand manner:

- Instructions to run your LibSVM code.
- The set of parameter values you used to obtain your SVM model.
- Your SVM model's performance in 5-fold cross validation. This *must* mention the accuracy for each fold separately, as well as the final accuracy (i.e., the average across all 5 folds).
- A short paragraph describing any time-related issues you ran into while trying our different parameters with LibSVM.
- Instructions to run your k-means code.
- The clustering performance, as described in Sec. 2.2.
- The set of parameter values you used to obtain the above performance.

[2] https://github.com/kronusme/dota2-api/tree/master/data has the details about the clusters, game modes, game types, and the heroes. They are for human understanding/interest only, and have no relevance to the algorithm.

[3] https://www.cs.waikato.ac.nz/~ml/weka/

[4] https://scikit-learn.org/stable/

3 What to submit?

A single `.zip` file containing the following:

1. Your code that creates the input data for LibSVM to run.
2. The model file provided by LibSVM.
3. Your Python (or Java) code to run k-means clustering using `scikit-learn` (or Weka).
4. Your final report (at most 2 pages using 11 pt font and 1.5 line spacing) as a PDF document.
5. Your solutions to the "theory" component of this assignment as another PDF document.