***Special Topics in Advance Databases- Project 1***

Topic: SVM

|  |  |
| --- | --- |
| Name: | **Abu Ayub Ansari Syed** |
| UTA ID: | 1001297330 |
| Course ID: | CSE 6339 |
| Email: | abuayubansari.syed@mavs.uta.edu |
|  | |
| Name: | **Hoda Moradi** |
| UTA ID: | 1001121261 |
| Course ID: | CSE 6339 |
| Email: | Hoda.moradi@mavs.uta.edu |

**1. Download C.J. Lin's LibSVM software**

* Downloaded from website: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
* Unzip here.
* Go to /matlab/ sub folder.

Installation:

* Firstly, we need MATLAB installed in the system.
* Next, download the add-ons "Support for MinGW-w64 compiler 4.9.2"
* Install the add-on to the MATLAB.
* Run the 'make.m' file in this sub folder.
* Select the Microsoft Complier C++ for installation.
* This will install the LIBSVM add-on to the MATLAB library.

Refer: /matlab/README

**2. a ) Use handwritten data provided.**

* Copied data set from Blackboard.
* Paste in text file.
* Import into excel sheet with required settings such as delimiter ',',
* Saved as 'Hand\_Written\_Data.csv'.

Refer: /matlab/Hand\_Written\_Data.csv

**2. b ) Pick two classes, say A and B. Use all 39 instance of each class**

* Data is chosen as per the input arguments.
* Proper displacement is taken to choose the data from the csv file.
* Alphabets are converted to number using indexing of the 'alpha.xlsx' file.

Refer: /matlab/alpha/xlsx

**3. Run linear SVM. Obtain support vectors (SV's) and bounded support vectors (BSV's).**

* My code is named: "STDSP1.m".
* Run on MATLAB R2016b console.
* Give user input for the program.
* Which is the Alphabets, such as : A and B, I and C.
* The output on console shows the successful implementation of Linear SVM.
* Sample outputs are also attached in the folder.(explained below)

Refer: /matlab/STDSP1.m or /matlab/results/Code.pdf

**4. Check KKT conditions for each SV/and BSV's.**

* We run the svm predict on the Training data.
* This is to achieve the decision values for the training data.
* The KKT condition is analysed to finding the alpha values.
* The corresponding equation explains the same: ai{yi(xi · w + b) - 1 + ?i} = 0
* If there is a no SV for the data, then the alpha value is zero.
* For those which have SV, we calculate the (ksi) >=0
* Therefore,(ksi)= 1- yi(xi · w + b);
* We can get the "xi · w + b" value from the decision value of the svm predict.

Refer: /matlab/STDSP1.m or /matlab/results/Code.pdf

**5. Repeat this for two pair of classes. (i) A vs B. (ii) I vs C**

* Achieved from the user arguments to the program.
* Results show the respective console information.
* Experiment has been conducted for both: A vs B and I vs C

Refer: /matlab/STDSP1.m or /matlab/results/Code.pdf

**Steps to Run:**

1. Go to folder libsvm-3.21/matlab

2. Complete the installation as mentioned above.

3. Run the *STDSP1.m* file.

4. Make sure the data files sure in the same folder.

5. Data files: "Hand\_Written\_Data.csv" and alpha.xlsx"

4. Give user inputs in console.

5. View console for results.

(optional)6. Put a breakpoint at the end of main function to view the variables and model in workspace.

**Results:**

1. libsvm-3.21/matlab/results

2. The console, variables, Xi values, and Alpha values of both the experiments are shown.

3. Code.pdf has the code of the program for the same.

4. Accuracy achieved is 97.3684%.

5. ksi values show that the KKT conditions hold.