ITCS 6156 Spring 2017
ASSIGNMENT : 4

SUPERVISED LEARNING

SUPPORT VECTOR MACHINES

Rahul Rachapalli
800968032
rrachapa@uncc.edu

Support Vector Machines

(Supervised Learning)

Support Vector Machines(SVM):

SVM is a supervised machine learning algorithm that is used to classify linearly separable and nonseparable data. The algorithm used margins to draw boundaries between the data ie., given a set of labelled training data, the algorithm outputs an optimal hyperplane that categorizes the data. Fig 1 shows the optimal hyperplane that the algorithm has formulated to classify the dataset in thick green line.

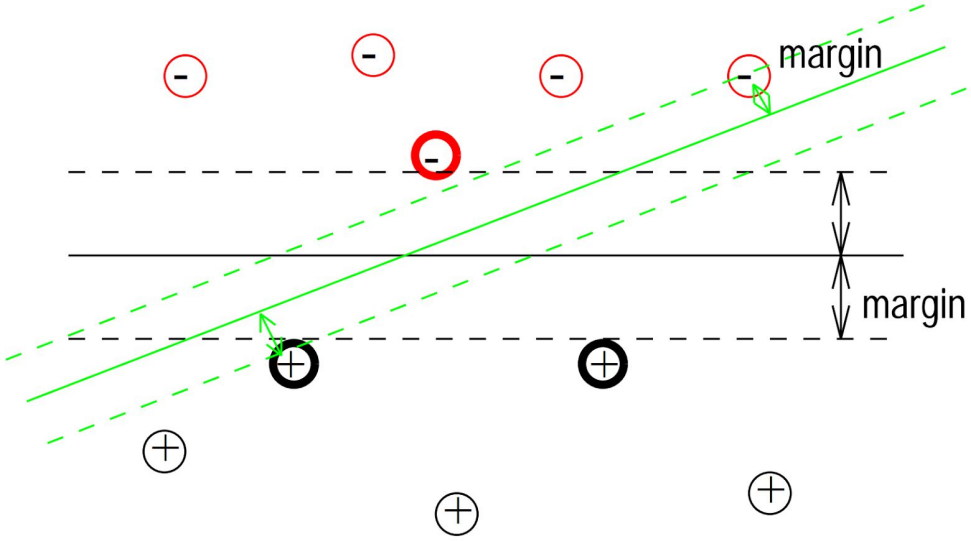

Fig 1 : SVM classifier

Support Vectors:

Support vectors are the data points that lie closest to the decision surface (or hyperplane) • They are the data points most difficult to classify. Yet, they help us to attain the maximum margin and thereby find the optimal hyperplane that classifies the data.

Margin:

For any data set, there exists multiple hyperplanes to classify the dataset. The algorithm chooses the optimal hyperplane by drawing two hyper planes that are the outliers in that dataset.

The algorithm draws two hyperplanes using the support vectors or Kernel points. Theses planes through the kernel points will have the largest margin than any other hyperplane in between the points as shown in Fig 2.

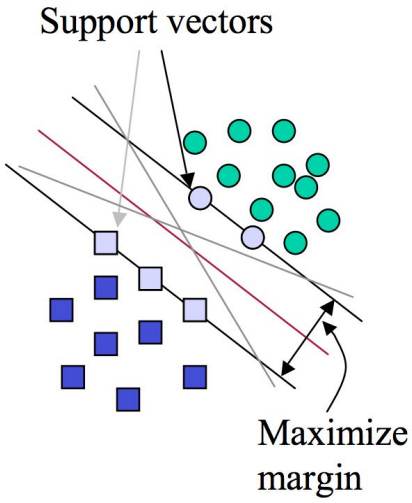

Fig 2: Support Vectors and Margin

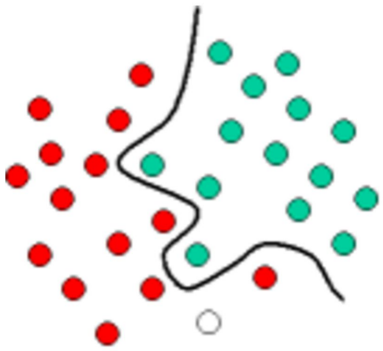

Fig 3: SVM Hyperplane for non separable Data

As shown in the fig 2, the optimal hyperplane is the one that lies equidistant from both the planes through the support vectors. Fig 3 shows the hyperplane that is formulated by SVM for a linearly non separable dataset.

Algorithm:

SVM is similar to Neural Nets where there are lots of training data that is represented. The output of is a set of weights for each feature whose linear combination is the predicted result for a given input set. The important difference is that we use the optimization of maximizing the margin to reduce the number of weights that are nonzero to just a few that correspond to the important features that 'matter' in deciding the separating line (hyperplane). These nonzero weights correspond to the support vectors (because they 'support' the separating hyperplane).

VC Dimension:

The VC dimension for the set of functions {f(α)} is defined as the maximum number of training points that can be shattered by {f(α)}.

Data Sets

A dataset is a collection of data represented in as a single statistical matrix. Given are two data sets

1. *Amazon reviews - Sentiment Analysis*
2. *Optical recognition of handwritten digits*

1. Amazon reviews sentiment analysis dataset

This dataset consists of reviews of Amazon products which is a subset of large Amazon review collection. The data set has 3 columns and the type of data is as follows:

- ➔ *Name of the product - String / text*
- ➔ *Review of the product - String / text*
- ➔ *Rating given by the user - Integer set {1,2,3,4,5}*

2. Optical recognition of handwritten digits

This data set consists of preprocessed normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13 to the test set. 32x32 bitmaps are divided into non-overlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates an input matrix of 8x8 where each element is an integer in the range 0..16. This reduces dimensionality and gives invariance to small distortions.

The dataset has 65 columns with 64 columns being the 8X8 matrix integer values and the last column being the digit that the 64 values represented.

Amazon Dataset Analysis and Prediction

Assumptions:

The name of the product is ignored as it is independent of the rating and ratings cannot be predicted using the name of the product. Hence, the rating depends on the reviews only.

Preprocessing:

Step 1: The reviews are separated and each review is converted into an array of words.

Step 2: Each word in the array is compared with stop words that are defined. If the word is present in the list of stop words, it is ignored and we proceed to the next word.

Step 3: If the word is not present in the list of stop words, we lemmatize the word and then stem it. At the end of step 3, we get a list of words that are lemmatized and stemmed.

Step 4: The words that are lemmatized and stemmed are converted into a single string and the word frequency is counted.

Step 5: The top 500 words or the whole list of words are considered as the attributes to dive the data.

Step 6: A sparse matrix is built based on the above attributes counting the occurrences of attributes in each review.

Stop Words:

The words 'the','a','an,'is','it' etc., are present in almost all the sentences and do not contribute to the decision process. All such words are called stop words and it is important to remove the stop words in pre-processing stage. So, we used some libraries like nltk to get a list of some pre-defined stop words.

Lemmatizing[2]:

The process of grouping together different forms of word to analyse it as a single word.

Stemming[3]:

The process of converting words into their stem or its root form. For sing, sang, sung, singing are represented in their base word form like sing*

SVM :

A hyperplane is used to classify the given data into different classes. The test data is then predicted based on the class it belongs to.

Prediction :

The rating of a review is predicted after the new review is pre-processed (lemmed and stemmed) and a sparse matrix is built for it using the same set of attributes that we decided to build the decision tree. The new sparse matrix is given as input to the k-NN algorithm that stores

the given matrix after converting it using ball_tree algorithm. The result is the predicted rating of the given review which will be done based on its k nearest neighbors.

Results:

There are 500 columns and all of them are used as attributes to predict the results. The prediction probability for this dataset is **59.5%** with **linear** kernel function and is irrespective of C.

NOTE : I was not able to run the whole amazon data set and the results here is for partial dataset only.

The classes that are used to classify the multi class data are

1. SVC
2. NuSVC
3. LinearSVC.

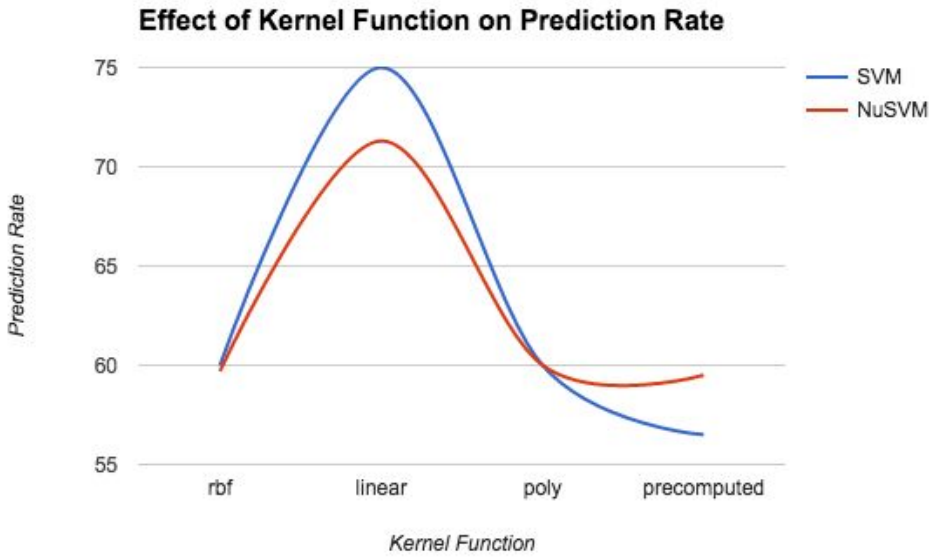

Fig 4 : Change in Prediction rate with Kernel Functions

Classifier/Kernel	rbf	linear	poly	precomputed
SVM	60	75	60	56.5
NuSVM	59.7	71.3	60	59.5

Table 1 : Dependency of Prediction rate on kernel function

Function	loss	c	max_iter	prediction rate
LinearSVC	squared_hinge	1	10000	76.2

LinearSVC	hinge	1	10000	75.2

Table 2 : Dependency of LinearSVC classifier on loss function

The effect of the number of C on various classifiers are as follows :

C value	Prediction Percentage
-10	59.9955
-5	59.9955
1	59.9955
5	59.9955
10	59.9955
20	59.9955
25 and above	59.9955

Table 3 : Dependency of C value on Prediction rate

From the above experiments, I have observed that the optimal kernel function is "**linear**" and is independent of C values.

It is also observed that the effect of weights

- uniform - assigning equal weights to all its neighbors
- Distance - emphasize importance of neighbors that are close by

Is negligible in our case.

Sample Run :

```
[rrachapa@mba-i1 src]$ pyt AmazonAnalysis.py
Setting up the system...
Importing Libraries
Defining grammar..
Preparing set of stop words
Analysing training dataset..
Loading the dataset...
Analysing testing dataset..
Loading the dataset...
Preprocessing the data set...
/users/rrachapa/anaconda2/lib/python2.7/site-packages/sklearn/utils/validation.py:526: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
Predicting...
------------------------------------------------------------


Prediction rate is 95.0
[rrachapa@mba-i1 src]$
```

Digit Recognition Dataset Analysis and Prediction

Assumptions:

There are no missing values in the dataset and the given data is accurate.

Preprocessing:

The data is arranged in the form of a list, read from the file and is represented as a Data Frame.

SVM :

A hyperplane is used to classify the given data into different classes. The test data is then predicted based on the class it belongs to.

Prediction :

A new dataset can be predicted using the data that has been classified. The only step that is needed for it to be predicted is that there are **64** columns in the given integer data set. The pre-processing stage allows us to represent this data in the form of a list. The new list is input to the decision tree for prediction.

Results:

There are 64 columns and all of them are used as attributes to predict the results. The prediction probability for this dataset is **97.4958%** with **polynomial** kernel function.

There are 2 algorithms to classify the multi class data are

1. SVM
2. NuSVM

The effect of the kernel function on various classifiers are as follows :

Classifier	Kernel	Prediction Percentage
SVM	rbf	54.09015025
	linear	95.9933222
	poly	97.49582638
NuSVM	rbf	11.01836394
	linear	96.04897051
	poly	97.49582638

Table 4 : Influence of Kernel function on Prediction rate

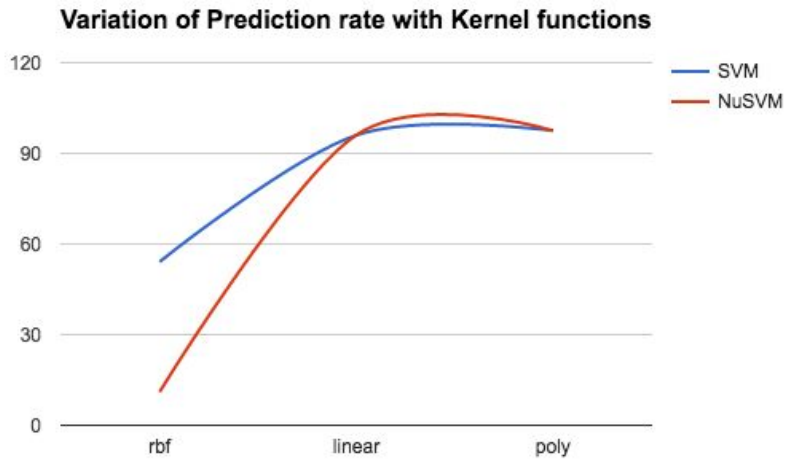

Fig 5 : Effect of Kernel function on Prediction rate

I have observed that the C value has not affected the prediction rate. It remained constant with varying C values for all the kernel functions.

C	Rbf	linear	poly
1	54.09015025	95.9933222	97.49582638
2	57.2064552	95.9933222	97.49582638
3	57.2064552	95.9933222	97.49582638
4	57.2064552	95.9933222	97.49582638
5	57.2064552	95.9933222	97.49582638
10	57.2064552	95.9933222	97.49582638
15	57.2064552	95.9933222	97.49582638
25	57.2064552	95.9933222	97.49582638
50	57.2064552	95.9933222	97.49582638
75	57.2064552	95.9933222	97.49582638
100	57.2064552	95.9933222	97.49582638
150	57.2064552	95.9933222	97.49582638
200	57.2064552	95.9933222	97.49582638
250	57.2064552	95.9933222	97.49582638
500	57.2064552	95.9933222	97.49582638

Table 5 : Effect of C value on Prediction rate

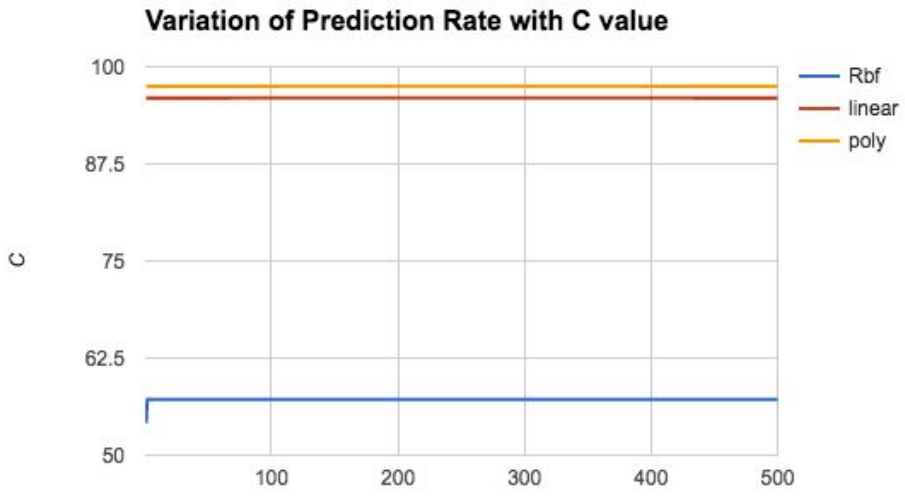

Fig 6 : Effect of C value on Prediction rate

Form the above experiments, it is observed that both polynomial and linear kernels give us the same prediction rate. However, I feel linear function gives us a better prediction to classify data than polynomial kernel. Hence, I have used linear kernel function with default parameters.

Sample Run:

```
[rrachapa@mba-i1 src]$ pyt opticalRecogniser.py
Setting up system
Importing libraries
Reading training dataset
Reading test dataset
Processing training dataset
Processing test dataset
Building a Neural Network from training data
Cross Validation score : [ 0.90697674  0.95121951  1.          0.97368421  0.86842105]
Prediction rate is 96.2715637173
[rrachapa@mba-i1 src]$
```

References

1. ITCS6156_SLProject.pdf
2. https://en.wikipedia.org/wiki/Lemmatisation
3. https://en.wikipedia.org/wiki/Stemming
4. http://documents.software.dell.com/Statistics/Textbook/Text-Mining
5. http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
6. http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
7. https://en.wikipedia.org/wiki/Support_vector_machine
8. http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf
9. http://www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro_svm_new.pdf
10. http://cs229.stanford.edu/notes/cs229-notes3.pdf
11. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/svmtutorial.pdf