ML Report Final

Navneet Agarwal - 140100090
C Vishvesh - 140050031
Sohum Dhar - 140070001
Tanmay Parekh - 140100011

April 2016

Contents

1 Description **2**
1.1 Description of the problem statement 2
1.2 Motivation 2

2 Approach and Implementation **2**
2.1 Approach towards the problem 2
2.1.1 Support Vector Machines 3
2.1.2 Perceptron Update 3
2.1.3 Neural Networks 3

3 Analysis **3**
3.1 SVM analysis 3
3.2 Perceptron analysis 5
3.3 NN analysis 5
3.4 Comparing SVM with Neural Network and perceptron 6

4 Future Work **7**
4.1 Improving the model 7

5 Conclusion **7**

6 Citations **7**

1 Description

1.1 Description of the problem statement

Malicious web sites are a cornerstone of Internet criminal activities. As a result, there has been a broad interest in developing system to prevent the end user from visiting such sites.We aim to explore various learning approaches for detecting malicious web sites using lexical and host-based features of the associated URLs.

1.2 Motivation

The detection of malicious urls can help keep users in a private network safe from external attacks from hackers, who launch attacks using malicious(e.g. phishing) web sites! The firewall can identify what url-requests should be allowed to pass and dropping the malicious requests.Users can even be attacked from hackers/attackers by viruses/autorun scripts being downloaded from a malicious url. These scripts can corrupt our machines or send valuable information to the attackers. We certainly don't want anything 'bad' to happen behind our back while we are online. Many times the network administrator wants to prevent request to certain kinds of web sites, deeming them as malicious. e.g. in China, sites on Tiananmen Square can be blocked in the network. or parental control. We can prevent attacks on cloud servers, internet-backbone-routers,servers . E.g. attacks where the cloud servers/backbone-routers are being requested to get a malicious web page, which carries hidden scripts which can bring down the server. Many times some malicious web sites track user browsing habits and sell this information to interested parties, e.g. for advertising. These use cookies, multiple advertisement links etc. to track the user! These sites can be blocked ensuring user privacy. Many a times the (naive) users are unaware of such activities on the internet!

2 Approach and Implementation

2.1 Approach towards the problem

We use the sparse dataset available at and try to understand the kind of feature space based on the description.We try different models on this dataset, based on understanding of the dataset and the feature space. We divide the dataset into training and validation to ensure that our model works well on new, unseen urls based on patterns in the url information.We apply the following three models based on what we have learnt in class.

- Support Vector Machine
- Perceptron update
- Neural Networks

2.1.1 Support Vector Machines

SVMs work well for classification problem, maximizing the margins to get a separating hyperplane with enough breathing space!We use the fact that SVMs are able to work with higher implicit dimensions and even handle sparse datasets as ours.

2.1.2 Perceptron Update

Perceptron update works well for classification problem, with a separating hyperplane. We use the feature-set and believe that there is separating hyperplane and run the algorithm for perceptron update algorithm for some number of iterations.However the perceptron hyperplane does not give enough breathing space for the dataset!

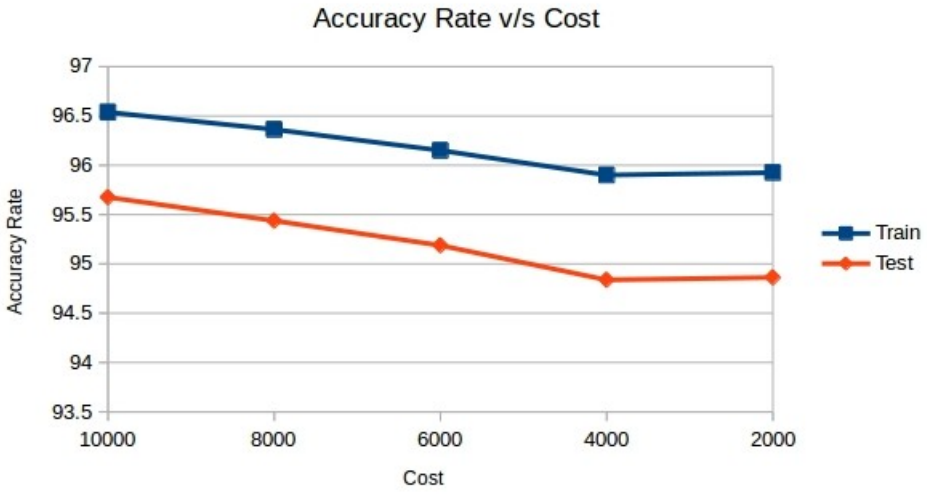
2.1.3 Neural Networks

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning.We implement our own neural network and tune various parameters like learning rate and number of hidden layers and number of units in each hidden layer etc. We implement back- propagation algorithm with squared error function (for simplicity).We use cross validation to analyse the results.Neural Network beats SVM!

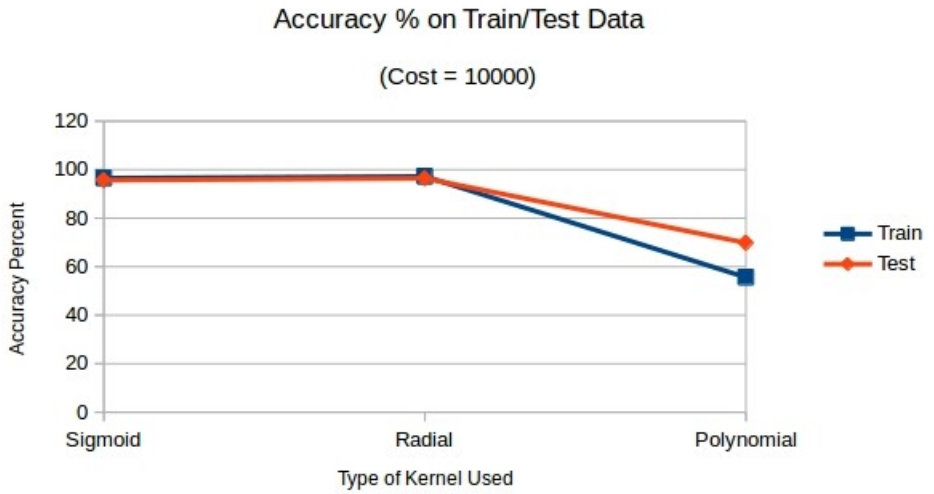
3 Analysis

3.1 SVM analysis

First we analyzed the relation between the accuracy of the model and the cost of the model. We have also included the accuracies for train and the test data. The below graph summarizes the results -

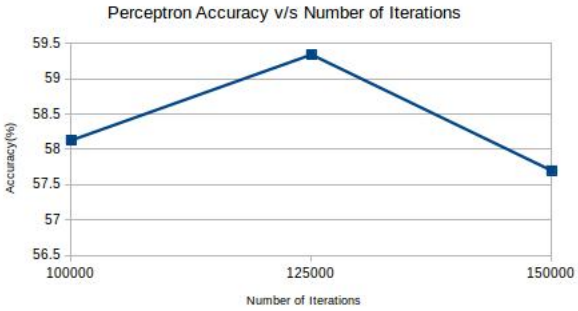


Then, we analyzed the change of kernel and it's effect on the accuracy of the model. We again took into account the train and the test accuracies. Here are the results -



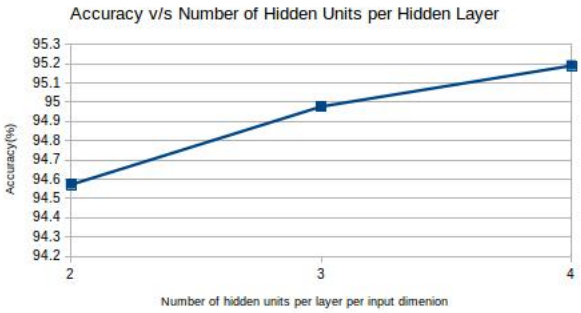
3.2 Perceptron analysis

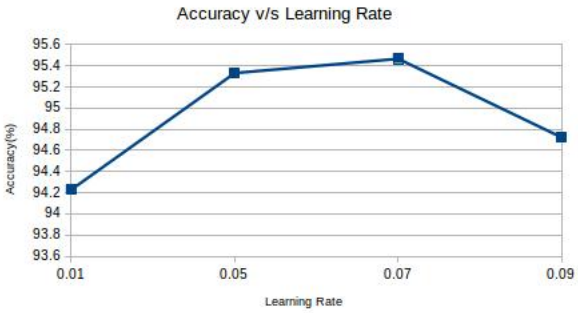
Here we analyze the Perceptron accuracy with number of iterations(threshold for algorithm)

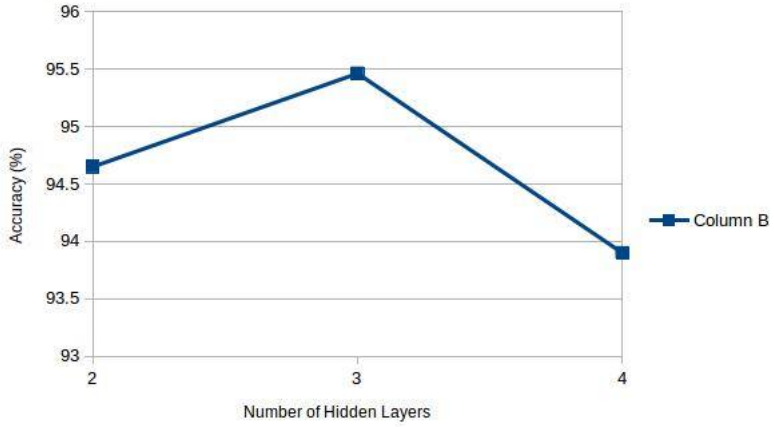


3.3 NN analysis

Now, we move on to Neural Network!
We analyze the variation of Accuracy of the model over the test dataset as we vary various parameters like the learning rate, the Number of hidden layers and the number of units in the hidden layer etc.







3.4 Comparing SVM with Neural Network and perceptron

Here are the tables for indicating true positives, false positives, etc.

This one is for SVM (94-95% accuracy)

	True	False
Positive	2391	388
Negative	5198	23

This one is for Perceptron (57-60% accuracy)

	True	False
Positive	429	1409
Negative	4177	1985

This one is for Neural Networks (95-96% accuracy)

	True	False
Positive	2316	280
Negative	5306	98

4 Future Work

4.1 Improving the model

We can analyse the feature set more with better understanding of the feature space. We can use better dimensionality reduction approaches for huge feature spaces(for NNs).We can use regularization for NNs to ensure our model works very well for unseen data.What would be better than implementing our model on a live network to test it in practice for random and real life test data!! Maybe we could implement our model on our personal wireless access point, just for fun.

5 Conclusion

It was a great experience working with different models learning and implementing new algorithms. Neural Networks are great models to work with!

6 Citations

- Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

- Justin Ma, Alex Kulesza, Mark Dredze, Koby Crammer, Lawrence K. Saul, and Fernando Pereira, Exploiting Feature Covariance in High-Dimensional Online Learning Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pages 493-500, Sardinia, Italy, May 2010.