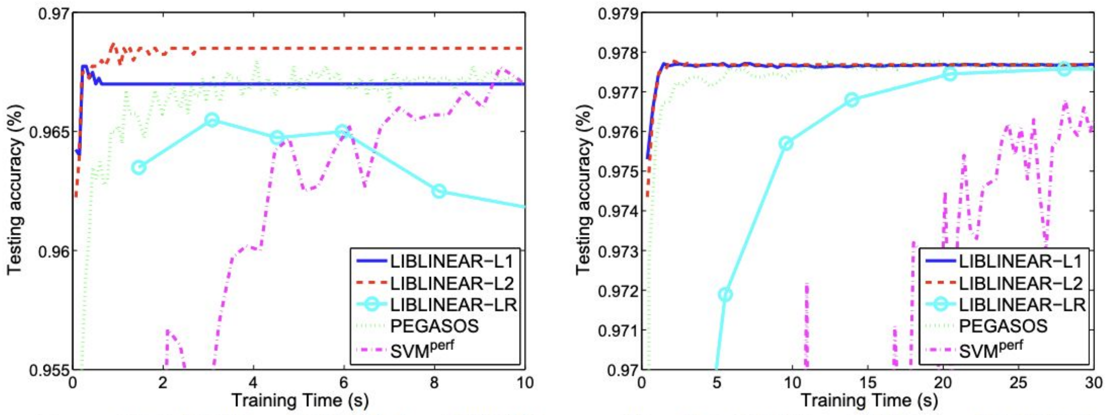
Austin Corum
CS599 - Machine Learning Research
November 1, 2019

Project 2 Week 1

Option 1:



(a) news20, l: 19,996, n: 1,355,191, #nz: 9,097,916   (b) rcv1, l: 677,399, n: 47,236, #nz: 156,436,656

Figure 1: Testing accuracy versus training time (in seconds). Data statistics are listed after the data set name. l: number of instances, n: number of features, #nz: number of nonzero feature values. We split each set to 4/5 training and 1/5 testing.

Figure Referenced from: Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C., LIBLINEAR: A Library for Large Linear Classification, Figure 1(a)

Problem Setting:
The input in these figures are key-value pairs, the output in this figure is the total accuracy of linear classifiers(SVM). In these figures, there are actually multiple functions to learn, which may make this figure difficult to reproduce. The functions to learn in this figure are L1-SVM, L2-SVM, LR, Pegasos, and SVMperf. For this figure, I am only going to cover the first 3, cause the paper does not clearly mention the algorithms for these L1-SVM solvers (they are referenced in another article, but are non-optimal).
Data Sources:
A real world example of this problem could consist of email spam. Given data that repeats, we can classify this data to distinguish what may actually be/not be spam. The dataset for this problem is given by the article with a link to the dataset.
(https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#news20 / https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#rcv1.multiclass)
The news20 dataset has a size 15,935 training data (rows), and 3,993 testing data(rows). Also, this dataset contains 62,000 testing/training features. The RCV1 dataset has a size of 15,564 training data (in rows), and 518,571 testing data (in rows). Also, this dataset contains 47,000

features. The data seems to be the input, X in this case, and the output Y is the accuracy of the algorithms on this data.

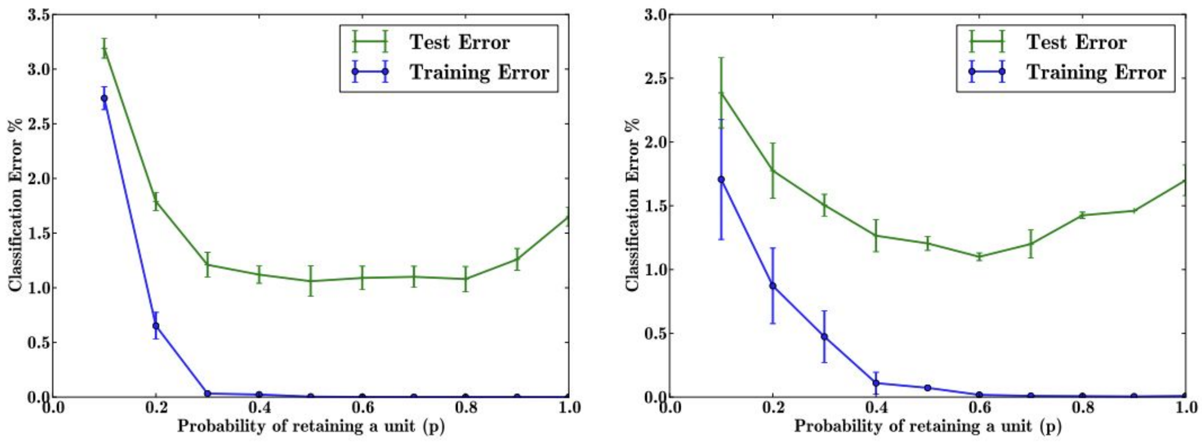
First Row of Data:

1 197:2 321:3 399:1 561:1 575:1 587:1 917:1 1563:1 1628:3 1893:1 3246:1 4754:2 6053:1 6820:1 6959:1 7571:1 7854:2 8407:1 11271:1 12886:1 13580:1 13588:1 13601:2 13916:1 14413:1 14641:1 15950:1 20506:1 20507:1

Algorithms:

In these figures, I will train the data and compare the trained data with the testing data to find out the overall accuracy. I will install/use the library LIBLINEAR to help with gaining the information from the data. I plan to run though all the algorithms and loop through each row to train the data in each function, then store their accuracy in a file to plot after the program has finished. If I chose this figure, I will learn how to utilize SVM libraries to classify data, as well as find out which SVM algorithm is optimal for specific data.

Option 2:



(a) Keeping n fixed. (b) Keeping pn fixed.

Figure 9: Effect of changing dropout rates on MNIST.

Figure Referenced from: Srivastava, N., Hinton, G.,Krizhevsky, A., Krizhevsky, I., Salakhutdinov, R., Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Figure 9

Problem Setting:
With limited training data, many complicated relationships between inputs/outputs will be a result of sampling noise. They will exist in the training set, but not in real test data even if it is drawn from the same distribution. This complication leads to overfitting, this is one of the algorithms to help prevent it from occurring. The input for this figure is a dataset of handwritten digits, and the output after adding dropout are different values that describe the outcome of applying the dropout method. All in all, less error is outcome after adding dropout.

Data Sources:
A real world problem that this can apply to is google searching, someone may be searching for a movie title but they might only be looking for images because they are more visual learners. So dropping out the textual parts, or brief explanations will help you focus on the image features. The article states where they retrieve the data from (http://yann.lecun.com/exdb/mnist/). Each image is a 28x28 digit representation. The y labels seem to be the image data columns.

Algorithm:
My goal in reproducing this figure is to test/train the data and calculate the classification error for each probability of p (probability of retaining a unit in the network). My goal is to get p to increase as the error goes down to show that my implementation is valid, and I will tune this hyperparameter to get the same outcome. I will do this by looping through all the training and testing data using a 784-2048-2048-2048-10 architecture and keep the n fixed then change pn to be fixed. I will then gather/write the data into a csv file. This csv file will then contain all the

necessary data to output the figures. In this project, I will learn how the dropout rate can benefit the overall error in a neural network.