Project 1 Writeup

Andrea Bajcsy Charles Parker

03/02/2016

WU1 The classification accuracy, `Acc`, can be expressed by

$$\texttt{Acc} = \frac{1}{N} \sum_{k=1}^{N} \texttt{[datasets.TennisData.Y(k) == h.predictAll(datasets.TennisData.X)(k)]}$$

$$= \texttt{mean (datasets.TennisData.Y == h.predictAll(datasets.TennisData.X))}$$

`datasets.TennisData.Y == h.predictAll(datasets.TennisData.X)` produces an array that consists of a 1 at the indices the arrays agree and a 0 elsewhere. The arrays agree only when the test label matches the prediction.

`datasets.TennisData.Y > 0` converts all of the negative labels to 0 and keeps the positive labels at 1. Call this new array `a`. Similarly, `h.predictAll(datasets.TennisData.X) > 0)` converts all of the predicted negative labels to 0 and keeps the predicted positive labels at 1. Denote this array as `b`.

`a == b` produces an array that consists of a 1 at the indices the array agree and a 0 elsewhere. The arrays agree only when the test label matches the prediction since the -1 labels have essentially been replaced by a 0 label. Therefore, `a == b` is the same array as `datasets.TennisData.Y == h.predictAll(datasets.TennisData.X)`, and the computations are equivalent.

WU2 Training accuracy tends to decrease because as the the number of input data points increases while tree height remains constant, the likelihood of misclassifying new data points increases. In other words, the tree is not able to learn any new features to separate a more diverse sample space.

The test accuracy is not monotonically increasing because the initially small samples of data are not sufficient to allow inference to the real data distribution. At a certain point during testing, the number of data points allows for a sufficiently accurate representation of the real data. In other words, the tree is unable to generalize from such a small sample size at first. Then, the sample size reaches a threshold that closely matches the true data distribution. Larger samples closely match the relative composition of this threshold sample size.

The jaggedness arises from the inability of a small data sample to sufficiently represent the true data distribution compared to a large sample. Some small samples, by chance, may accurately represent the real distribution, while others, by chance, do not. For this reason, some relatively small samples allow for higher test accuracy while the others have a low test accuracy, causing the graph to exhibit erratic behavior for small samples.

WU3 We are guaranteed to see training accuracy monotonically increasing as the tree gets deeper because we are considering more features to partition the data. On the other hand, we expect that test accuracy will increase and then start to decrease in a hill-like fashion due to overfitting.

WU4 Overfitting does not appear to occur for the values of K tested. This is apparent from the presence of positive slopes for larger sample sizes in Figure 1. Therefore, underfitting is occurring the entire time. However, for $\epsilon \geq 10$, overfitting occurs. For $\epsilon = 10$, overfitting occurs after about 50 samples, as the test accuracy decreases. For $\epsilon = 15, 20$, an even sharper decrease in test accuracy occurs after about 10 samples. These decreases can be seen in Figure 2.

WU5 The results almost mimic those for random data. In higher dimensions, the distances between points is more concentrated in a small range, as shown in 3. In lower dimensions, the distances between points is more spread out, demonstrated in 4. The exception here is 2 dimensions, shown with the red outline. Many of the points are very close to each other with respect to only 2 features (in fact, they are 0 distance apart). This is not necessarily surprising since the two features chosen at random are probably black pixels on most images. 8 dimensions, outlined in brown, is also more spread out than on random data. One possible explanation is that pixels are not distributed randomly throughout the image, rather most non-zero pixels are concentrated towards the center of the image. Otherwise, the data behaves as expected.

WU6 The learning curve for the perceptron with 5 epochs is shown in Figure 5. The impact of the number of epochs on train/test accuracy is shown in Figure 6.

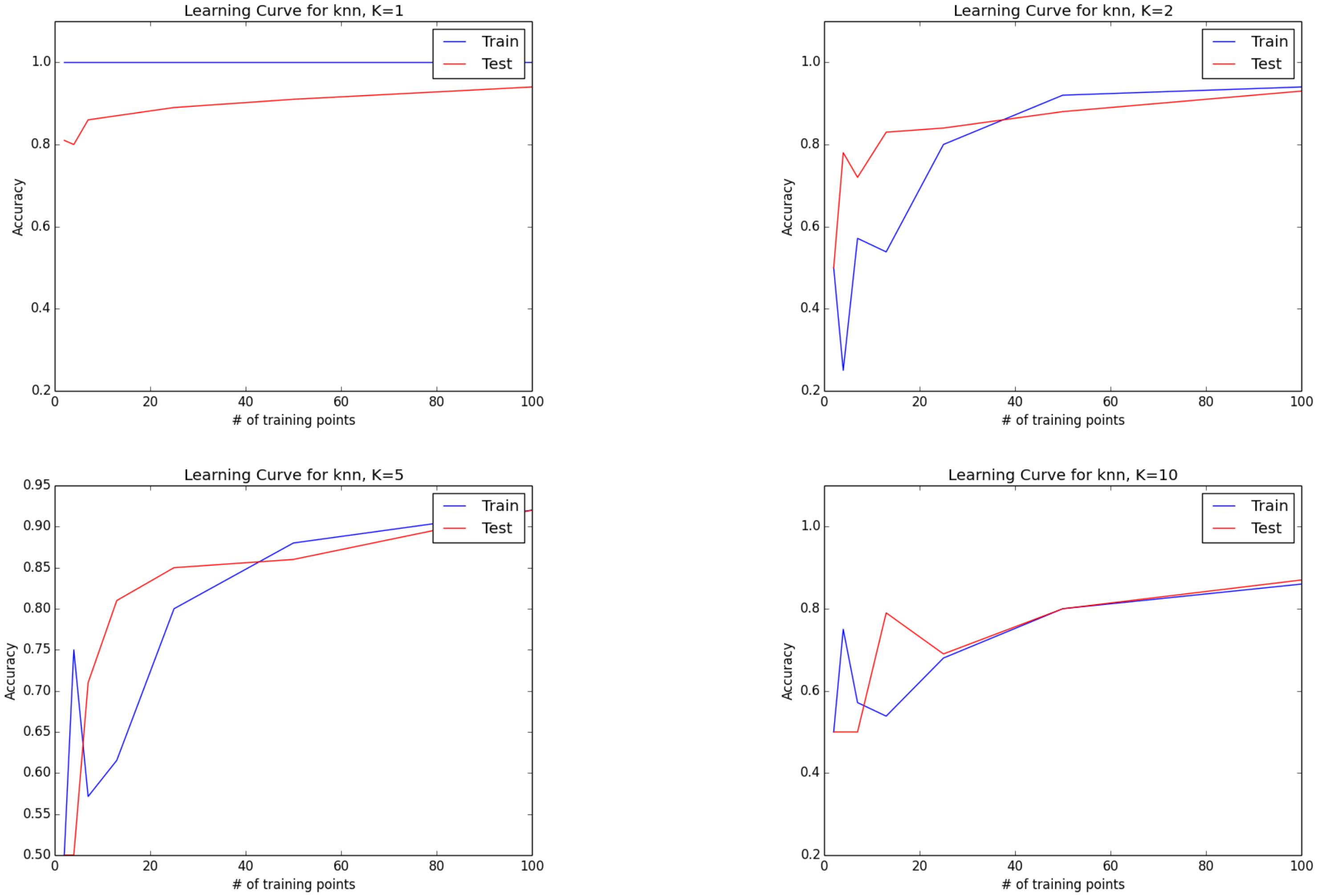

Figure 1: Learning curves for various K values

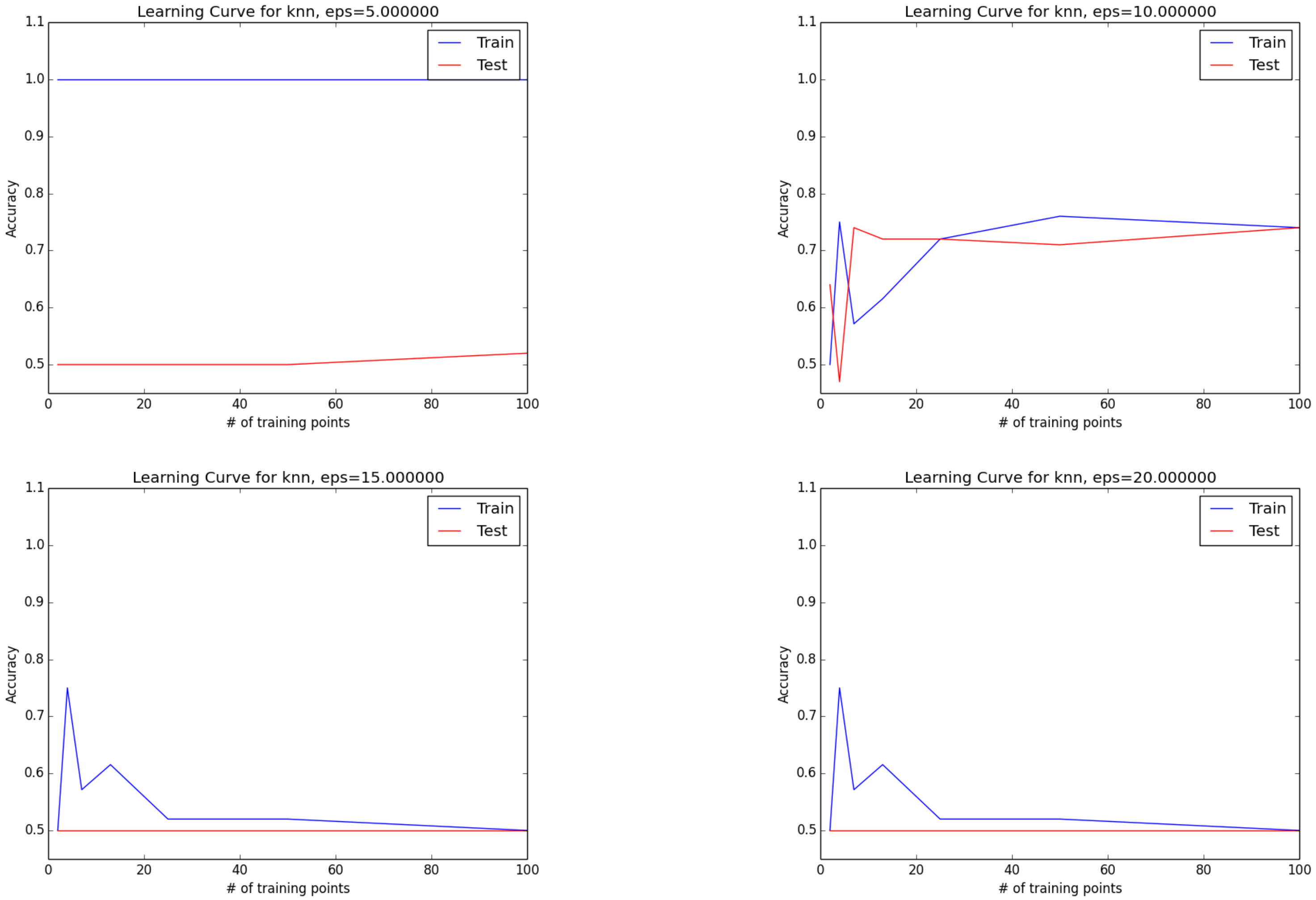

Figure 2: Learning curves for various ϵ values

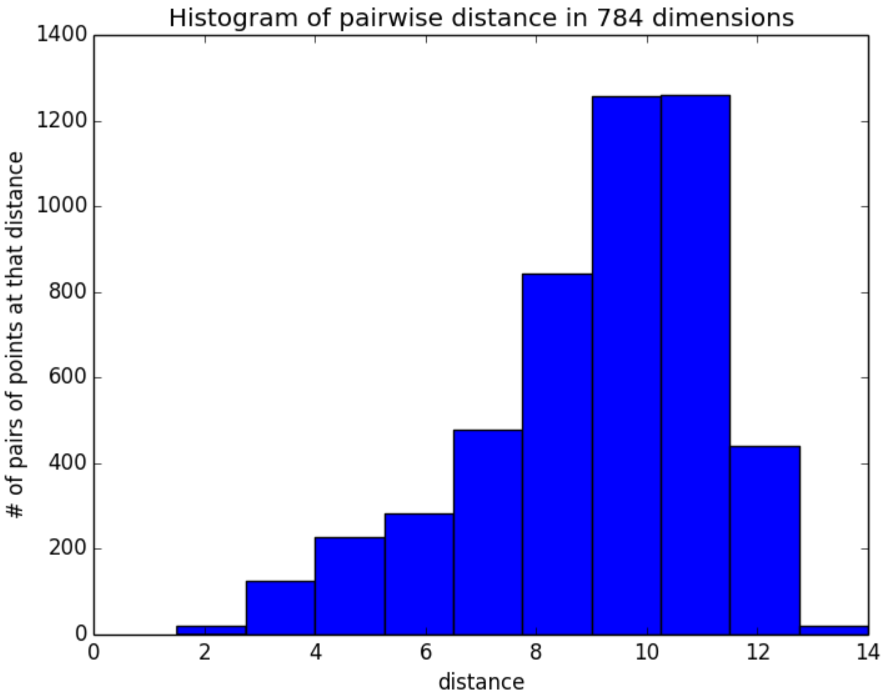

Figure 3: Pairwise distances using all 784 dimensions

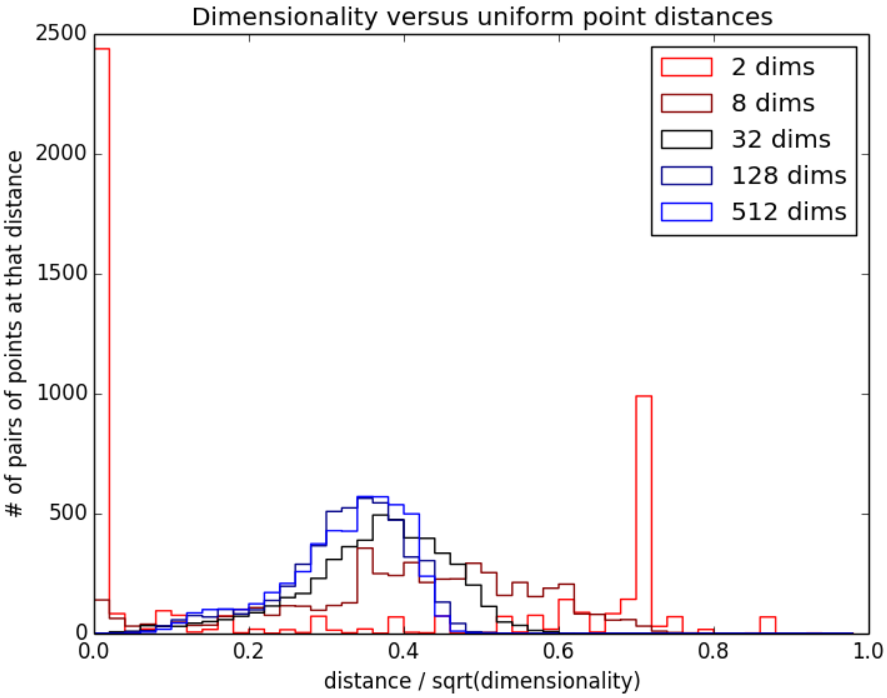

Figure 4: Pairwise distances using subsampled dimensions

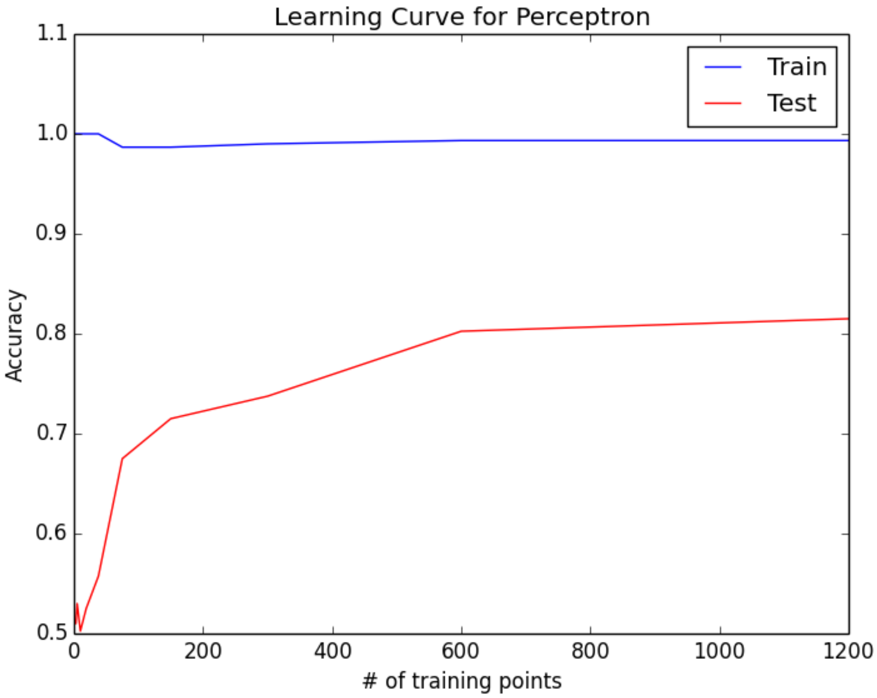

Figure 5: Learning curve for perceptron using 5 epochs

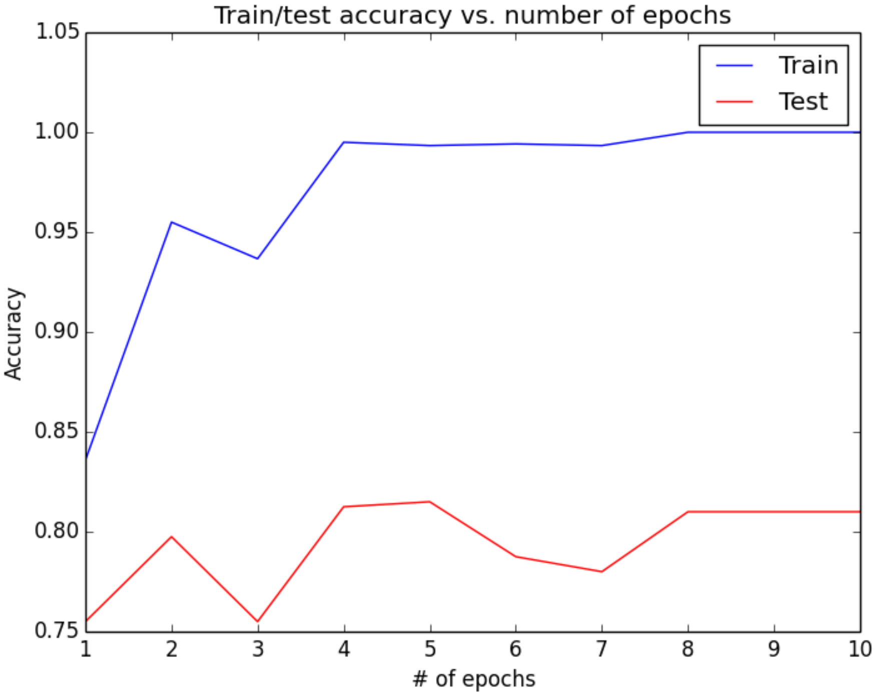

Figure 6: Effect of number of epochs on test/train accuracy