Introduction

The first section of this project will be re-implementing the paper referenced in [1]. This paper evaluates the performance of different norms as distance metrics in the hope of proving that lower order norms perform better on high dimensional data. They provide theoretical and empirical results to develop and support this argument. We will be carrying out simulations to prove their theoretical results and will be applying these results to a new algorithm. The structure of this report will be as such: we will give a high-level explanation of the paper, then present results from our simulations and finally apply this intuition to a new algorithm.

Motivation

Many algorithms use the Euclidean distance metric (L2 norm) in high dimensional problems as a natural extension of its use in two or three-dimensional space. It is unintuitive for us to think about a difference in distance metric performance in higher dimensions as we are only able to visualize a three-dimensional world. This is an instance of the curse of dimensionality. The authors of [1] propose and investigate the use of the Manhattan distance (L1 norm) and other lower order norms for calculating distance in high dimensions. They provide theoretical proofs and empirical results that show that as the dimension increases, lower order norms perform better.

Methodology

The distance metrics considered were:

Between two points $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$

Euclidean Distance :

$$L_2 = (\sum_{i=1}^{n}(x_i - y_i)^2)^{1/2}$$

Manhattan Distance :

$$L_1 = \sum_{i=1}^{n} | x_i - y_i|$$

Minkowski Distance :

$$L_p = (\sum_{i=1}^{n} | x_i - y_i |^p)^{1/p}$$

The Minkowski distance is a generalization of the Euclidean distance and Manhattan distance where p is typically greater than or equal to 1. Fractional values of p were also and we refer to these as *fractional distances or fractional norms*. Distance metrics were evaluated based on how well they discriminate between a given point and a query point. The query point in our analysis is the origin.

The pairwise distance between points in a simulated dataset was first investigated. We sampled from the Uniform (0,1) and Normal (0,1) distributions. These distributions were used because they

are easy to work with and have simple properties. 500 points were sampled from each distribution and the dimensionality of the dataset was varied (for example: sample 500 4-dimensional points, sample 500 50-dimensional points, etc.). The pairwise distances between every point was then calculated using a given distance metric and density curves were plotted. This was done to get an idea of the pairwise distance as dimensions increase.
Next, we turned our attention to investigating the convergence of the relative distance ratio, which will be introduced later, by comparing the distance between a point in a dataset and its nearest and furthest neighbors.

Results

Below are the results from simulations using the Uniform (0,1) distribution. Figure 1 shows a plot of the densities of the pairwise distances. As the dimension of the points increase, the pairwise distance also increases. Figure 2 shows a plot of the *average* pairwise distance versus number of dimensions. This plot interestingly shows a nonlinear relationship between the average pairwise distance and dimension. It also seems as though the average pairwise distance is converging to some value, indicating a breakdown of the distance metric in very high dimensions.

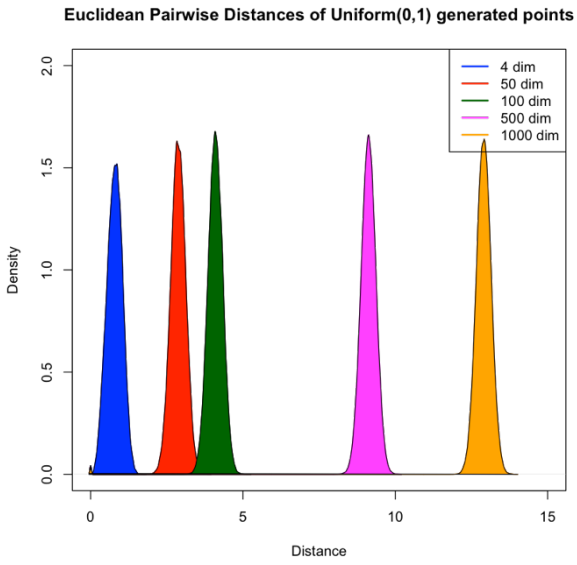


Figure 1

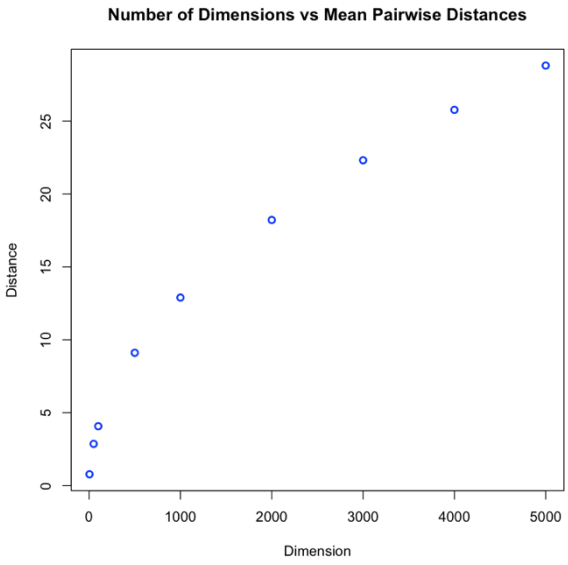


Figure 2

The same simulations were performed on data simulated from a Normal (0,1) distribution and the results are shown in figures 3 and 4 below. Similar trends are observed, except the pairwise distances themselves are larger. This is simply because the range of values of a normal distribution is larger than that of a uniform distribution.

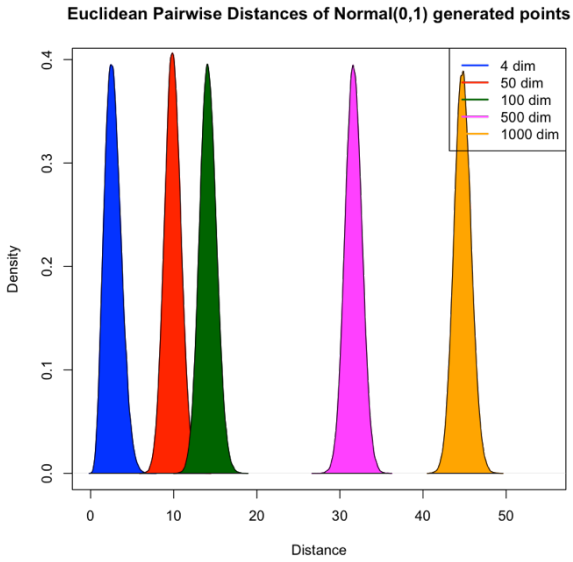


Figure 3

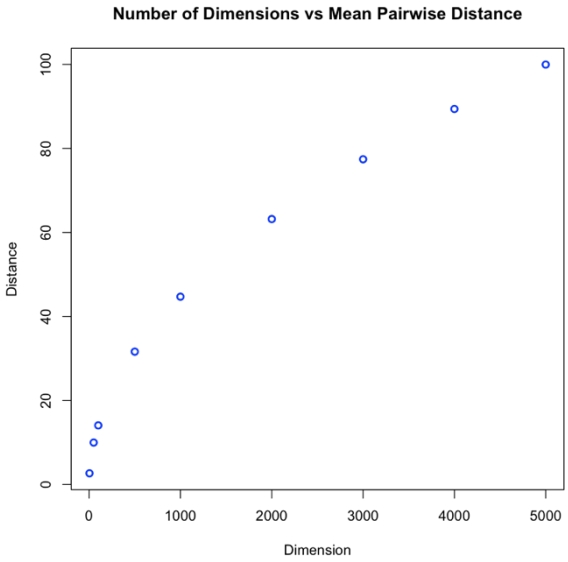
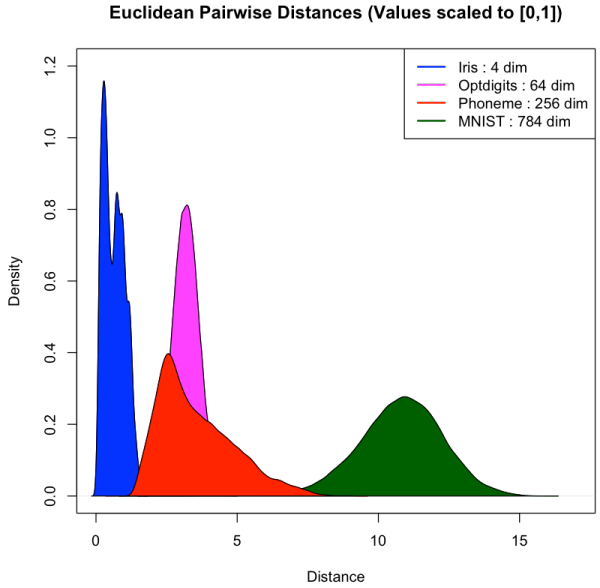


Figure 4

Due to the potential gap between theoretical performance and real world performance, we decided to also run this analysis on real-world datasets. Datasets of varying dimensions were chosen and were as follows: Iris flower data set, handwritten letters, handwritten digits, and a phonetics dataset. This analysis was just to show that real-world data may not always conform to theoretical results.



In high dimensions, data becomes sparse, causing the distance between a given point in a dataset and its nearest and furthest neighbor to converge to the same value. This compromises the accuracy of distance-based tools, for example: nearest neighbor search, k-nearest neighbors and k-means clustering. These methods assume that points that are close together in space are similar and points that are far apart are different. Since the sparsity of the data makes the distinction between close and far points difficult, these algorithms start to perform poorly.

For this simulation, 500 points were generated from a Uniform (0,1) distribution. For each point, the distance of the farthest point, D_{max}, and distance of the closest point, D_{min}, was calculated. The Euclidean distance, Manhattan distance, Minkowski distance with $p = 0.5$ and with $p = 0.1$ were chosen as distance metrics. The ratio shown below was used to measure the performance of each metric.

From figure 2 and 4, we saw that there is a possible convergence of pairwise distances in high dimensions. If this holds true, then $D_{max}$and D_{min} will converge to be the same value and their ratio will converge to 1.

$$Relative\ Distance\ Ratio = \frac{D_{max}}{D_{min}}$$

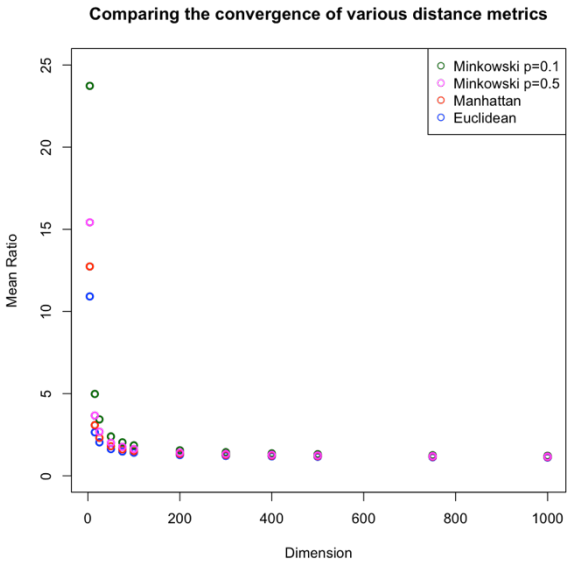


Figure 5

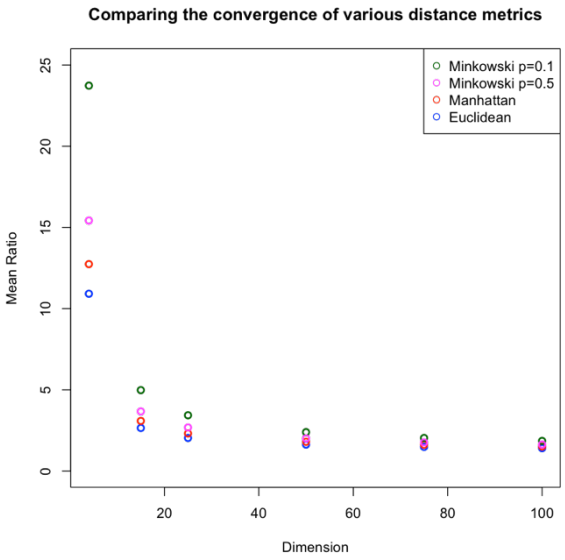


Figure 6

Figure 5 and 6 show plots of the mean relative distance ratio vs the number of dimensions of the points. From these graphs, we can see that for all distance metrics, the mean ratio converges to 1 in very high dimensions. Figure 5 shows the mean ratio for dimensions 1-1000 and it can be noted that after about the 200th dimension all metrics start to perform in a similar manner with the mean ratio converging to 1. Figure 6 shows dimensions 1-100 to give a closer look at the behavior of these metrics in lower dimensions. From this graph, the Minkowski distance with $p = 0.1$ converges the slowest, followed by Minkowski distance with $p = 0.5$, then the Manhattan distance, and finally the Euclidean distance. These simulations show that the L_p norms with smaller p values breakdown slower, in the sense that the curse of dimensionality affects them less.

Applications of Fractional Norms

Both L1 and L2 norms are used for regularization, however, only the L1 norm enforces sparsity in model parameters. We were curious as to how $L_{p<1}$ norms would perform in comparison to the L1 norm. Due to the non-convexity of $L_{p<1}$ norms ($L_{p<1}$ is mathematically not a norm, however we

will refer to it as such), we would have to settle for some local minima as opposed to the global minima as in L1 norm regularization. This encouraged us to implement $L_{p<1}$ norm regularization on neural networks as these local minima should not be as costly as in a regression setting.
We were confident that in a neural network setting, the $L_{p<1}$ norm would perform better than the L1 norm given that sparsity is encouraged with high dimensional data and the previous sections showed the superior performance of $L_{p<1}$ norms over the L1 norm on high dimensional data.
Unfortunately, fractional norms are not differentiable at 0 which would make it difficult to be implemented as we would not be able to use the backpropagation algorithm. [2] proposes a linear variation approximation that would approximate this non-differentiable function to a new one that can be solved using typical LASSO regression methods. However, this paper proposed a solution to the implementation of these regularization functions in a logistic regression setting, meaning that in a neural network setting, we would not be able to take advantage of the backpropagation algorithm.
To overcome this, we used the method of implementing L1 regularization in [3] and set the value of the derivative of the norm at 0, to 0, to overcome the discontinuity. The new optimization problem became:

$$\min\{(y - \hat{y})^2 + |w|^p\}, p < 1$$

The backpropagation algorithm was used to differentiate $(y - \hat{y})^2$ while $|w|^p$ was differentiated to be $\frac{p|w|^p}{w}$ and 0 whenever $w = 0$. The intuition behind setting this derivative to 0 is because if $w = 0$, it cannot be reduced anymore.

To test the suitability of fractional norms for regularization, the "ionosphere" dataset provided in Assignment 3 was used. This dataset has 33 features, which we thought was ideal for this comparison as extremely high dimensional data will cause all norms to behave in a similar way. A 1-layer feed forward neural network was fit to this data and train and misclassification error was compared for each of the 4 norms seen in the previous sections. In each test, the neural network had 7 nodes in the hidden layer and was fit with 4000 epochs and a regularization parameter of 0.006. The results are shown below:

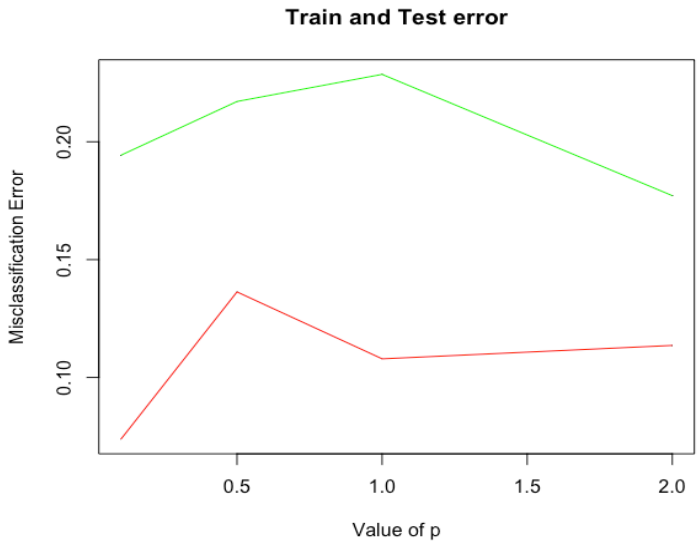


Figure 7

Figure 7 shows the misclassification rate versus the value of the norm for the test error (green line) and train error (red line). Interestingly, the smallest value of p for the L_p norm, did not perform as well as the L2 norm. We believe this is due to the approximation of the differential and lack of convexity. However, the $L_{p<1}$norms did perform better than the L_1 norm, further, the $L_{0.1}$ norm performed the best of these and the second best overall. Below is a table showing the training and testing misclassification errors for each norm:

	0.1	0.5	1	2
train.err	0.0738	0.1363	0.1079	0.1136
test.err	0.1943	0.2171	0.2286	0.1771

Discussion

Although the intuition behind the use of fractional norms as regularization functions was sound, there was a fundamental flaw in it. Regularization is used in a regression setting to enforce sparsity and make models more interpretable. In a neural network setting, less weight is put on parameter interpretation and more weight is given to making accurate predictions. Given only this, sparsity would not be useful in a neural network setting, however, sparsity in model parameters can make some nodes in the hidden layer useless, as they are set to 0. This would make the selection of number of hidden layer nodes easier.

Contrary to our hypothesis before running these experiments, we found that the L2 norm provides the best regularization of model parameters, by having a lower test error and a closer difference in training and testing error, which is also very important. As for inducing sparsity, none of the model parameters were set to 0. This prompted us to conclude that fractional norms are not a suitable regularization function for neural networks.

One shortcoming of our project was not implementing these fractional metrics on distance metric learning algorithms. We proposed this idea to Professor Ghodsi, however, due to the complexity of the new objective function, we agreed it would be too difficult under the time constraints to move forward with this.

Conclusion

In conclusion, with the number of papers read and hours spent laboring over possible uses of non-differentiable functions, we can safely say that, not only have we concluded that the use of fractional norms in neural network regularization does not perform as well as we thought, but we have also learned a great deal of extremely interesting research pertaining to supervised and unsupervised learning. We would like to thank Professor Ghodsi and all the Stat441 staff for an amazing opportunity for independent learning and a great course.

References

[1] On the Surprising Behavior of Distance Metrics in High Dimensional Spaces, Charu C. Aggarwal, Alexander Hinneburg , Daniel A. Keim

[2] Learning with $L_{q<1}$ vs L_1-Norm Regularization with Exponentially Many Irrelevant Features, Ata Kaban, Robert J. Durrant

[3] Neural Networks and Deep Learning – Improving the wat neural networks learn, online book

[4] Stat 441 Course Wiki