BEHAVIOR OF DISTANCE METRICS IN HIGH DIMENSIONAL SPACES

ALEXANDER WOOD
THE GRADUATE CENTER, CUNY

Abstract. When analyzing distances between points in high dimensional spaces, the data points become sparse, massively decreasing the efficiency of analyzation techniques such as the nearest neighbor rule and clustering. In this paper, we use experimental methods to verify that in high-dimensional spaces, the distance between the closest and furthest points from a uniform distribution becomes similar, and show that the classification accuracy can vary wildly for perturbed data. Next, we show that even points generated from the Gaussian distribution fall prey to the curse of dimensionality, becoming sparse in high dimensions. We look at the distance metrics L^1, the city block metric; the metric L^2, the Euclidean metric; and L^∞, the maximum distance metric.

1. Introduction

A classical problem addressed by machine learning is that of finding the nearest neighbor to a point. The applications of this are vast, including data mining, pattern recognition, machine learning, the nearest-neighbor method, similarity indexing structures, image completion, and data compression [1], [2], [10]. The larger the dimension of a space, the more information we receive about each point. It would be easy to assume that with more information, we are able to gain more information about the distances between points as the dimension increases. However, we find that exactly the opposite happens! As Aggarwal and Reddy explain in [2]:

> conventional distance measures were generally designed for many kinds of low-dimensional spatial applications which are not suitable for the high-dimensional case. While high-dimensional data contain more information, they are also more complex. Therefore, naive methods will do worse with increasing dimensionality because of the noise effects of locally irrelevant attributes.

In high dimensional spaces, things which seem intuitive in one- or two-dimensional spaces become complex and bizarre. For example, the volume of a high dimensional sphere goes to zero as the dimension increases. However, for a sphere of radius R and a sphere of radius $1.1R$, the ratio of the volume formed between

Date: December 10, 2015.

these spheres goes to 1 as the dimension N increases [7]. In this way we see that the volume of the high-dimensional sphere is hugging the boundary. The larger the dimension of the space, the closer together the distance between the nearest neighbor and the farthest neighbor becomes [3]. This problem of counterintuitive behaviors in high-dimensional spaces was coined by Richard Bellman as "the curse of high dimensionality" [4].

In addition to the poor performance seen in high dimensional spaces with the distance between the nearest and farthest neighbors, we also run into problems with classification. Classification problems appear in a broad range of applications, including web-document classification and discrimination between cancerous and non-cancerous cells. However, the curse of dimensionality remains, and we see that standard classification methods in high dimensions retain poor performance [12].

In this paper, we first provide a background of technical results in the field and define the terminology in the paper. We next provide experimental results, where we run three experiments:

(1) First we analyze the nearest neighbor problem in high dimensions to empirically observe these theoretical results. We choose a number of points, K, and run the experiment on a range of dimensions, N. We observe the average ratio between the nearest neighbor and farthest neighbor from a point as the dimension N increases.
(2) Next we run an experiment on classification accuracy in high dimensions. We generate a set of $\mathcal{K}$ of K points and a set $\mathcal{M}$ of M points, randomly assign each point $\mathcal{M}$ in a class, and set the true class of each point in $\mathcal{K}$ to the class of its nearest neighbor in $\mathcal{M}$. We then perturb the points in $\mathcal{M}$, set the assigned classes of the points in $\mathcal{K}$ similarly, and observe the relation between the true classes and assigned classes of the data.
(3) Finally, we perform an analysis of the nearest neighbor problem in high dimensions on random points sampled from the normal distribution. This requires normalization of our data to the unit inteval [5]. Again, we choose a number of points, K, and run the experiment on a range of dimensions, N. We observe what happens to the value of the average ratio between the nearest neighbor and farthest neighbor to a point as the dimension N increases and note that when vectors whose components are independent$N(0, 1)$ are divided by the L_2 length, they become uniformly distributed on the unit hypersphere.

Lastly we have a summary and conclusion of all of these results.

Appendix A contains graphs of the ratio between nearest neighbor and farthest neighbor in dimension N when our data points have been drawn from the uniform distribution, the results from the first experiment. Appendix B contains graphs of the accuracy rate of the assigned classification, the results from the second experiment. Appendix C contains graphs of the ratio between nearest neighbor

and furthest neighbor in dimension N when our data points have been randomly drawn from the standard distribution, the results from the third experiment.

2. Technical

The nearest-neighbor method uses a set of training data points with assigned classes, along with a distance metric to assign classes to new data points. More concretely, in the nearest-neighbor method we have a measurement space X, a distance metric d on X, and a set of classes $\{c_1, c_2, \ldots, c_n\}$. We take a set of training data, which consists of pairs of points and classes $\langle (x_1, c_1), \ldots, (x_N, c_N) \rangle$, where the c_i are not necessarily unique. The nearest-neighbor rule takes a new point, x, and assigns it to the class c_m such that the following holds:

$$d(x, x_m) \leq d(x, x_i)$$

for all i [8].

We have our choice of distance metrics for this method, the choice of which depends on what we are trying to observe with our data and the meaningfulness of the metric in our measurement space. In this paper, we focus on three metrics: the city block metric, the Euclidean metric, and the maximum distance metric.

Let X be an N-dimensional measurement space, and let $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be points in X. The city block metric, $L^1 : X \times X \to \mathbb{R}$, is given as follows:

$$L^1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} |x_i - y_i|.$$

The Euclidean metric, $L^2 : X \times X \to \mathbb{R}$, is given by

$$L^2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

and the maximum distance metric, L^∞, is given by

$$L^\infty(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq N} \{|x_i - y_i|\}.$$

Because of the sparseness of data in high-dimensional spaces, our normal methods for analyzing this data become meaningless in high dimensions, and results for the nearest-neighbor method are one clear example of this phenomenon [11]. The nearest-neighbor rule relies upon our selection of a distance metric, increasing the importance of the behavior of these metrics in high dimensions. It is important that we select the metric for analyzing our data in a way that gives us a meaningful outcome. In low dimensions, the Euclidean metric is standard as giving the most meaningful results on distances between points. However, due to the curse of dimensionality, it is important that we run tests in high dimensions to determine what the most meaningful metric will be in these high dimensions.

Theoretical results confirm the curse of dimensionality. The L^k norm is subject to the curse of dimensionality for all values of k, including the metrics we are studying, $k = 1, 2$, and ∞. As discussed by Aggarwal, Hinneburg, and Keim, the meaningfulness of the values given by the distance metrics L^k worsens as k increases. More specifically, in a space of dimension N under distance metric L^k, the difference between the point with maximum distance and the point with minimum distance increases at a rate of $N^{1/k-1/2}$. This holds regardless of the distribution we choose for our data points. We see that when $k = 1$ this diverges to infinity; when $k = 2$ this is bounded by 1; and for $k = \infty$ this converges down to 0. Therefore, in high dimensions, the L^1 norm is preferable to the L^2 norm, which is preferable to the L^∞ norm. All in all, the L^k metric provides the most meaningful results for low dimensions, and quickly looses meaningfulness for dimensions greater than or equal to 3 [1]. Aggarwal, Hinneburg, and Keim conclude that

> for dimensionalities of 20 or higher it is clear that the Manhattan distance metric provides a significantly higher relative contrast than the Euclidean distance metric with very high probability.

In empirical experiments, we expect these trends to hold. If these theoretical results hold true empirically, the information we receive from a nearest neighbor analysis in high dimensions will lack meaning. Instead, we would see a need to modify the methods with which we analyze information which has high dimensions. Therefore, it is important that we run empirical analyses to understand the curse of dimensionality in high dimensions on a practical level.

In addition to measuring distance, and important aspect of the nearest-neighbor method is that of classification. Consider a set of classes $\{c_1, c_2, \ldots, c_\ell\}$. Let $P_{TA}(c_i, c_j)$ denote the probability that the true class of a data point is c_i and the class assigned by a decision rule is c_j. The confusion matrix is given as follows [6]:

		Assigned Class					
		c_1	c_2	$\cdots$	c_j	$\cdots$	c_ℓ
True Class	c_1						
	c_2						
	$\vdots$			$\ddots$	$\vdots$	$\iddots$	
	c_i			$\cdots$	$P_{TA}(c_i, c_j)$	$\cdots$	
	$\vdots$			$\iddots$	$\vdots$	$\ddots$	
	c_ℓ						

The probability of correct identification, P_C, occurs by adding up the diagonal entries of the confusion matrix [6]:

$$P_C = \sum_{i=i}^{\ell} P_{TA}(c_i, c_i).$$

Due to the curse of dimensionality,

> an increasing noise from irrelevant attributes may cause errors in the distance representation, so that it no longer properly represents the *intrinsic distance* between data objects [2].

Because of this, we will expect to see that the nearest neighbor method, and hence our assignment of classes, becomes meaningless due to the lack of information given to us by our distance metric of choice.

Consider now the Gaussian distribution. The fraction of a random sample of points which falls into a sphere of radius 1.65 is 90% in dimension one. However, as the dimension n increases, this fraction decreases, and is all the way down to below 1% when the dimension is greater than ten [8]. We see that in N dimensions, the Gaussian has density function

$$p(x) = \frac{1}{(2\pi)^{N/2}\sigma^N} \exp\left(-\frac{|x|^2}{2\sigma^2}\right)$$

where σ is the variance [9]. When $\sigma = 1$, within the unit sphere, integrating the probability density function will yield very little mass [9]. Therefore, even with N-dimensional points sampled from the Gaussian distribution, we run into the problem of sparseness of our data points.

3. Experimental Results

3.1. **Maximum and Minimum Distance Experiment.** Experimental results on high-dimensional spaces further verify the aforementioned "curse of dimensionality." First, consider a collection $\mathcal{K}$ of K N-dimensional points. Let $D_{min}(\mathbf{x})$ be the distance of the closest point from $\mathbf{x}$ in our collection of K points under distance metric D, and let $D_{max}(\mathbf{x})$ be the distance of the farthest point. We want to analyze the ratio

$$r(\mathbf{x}) = \frac{D_{min}(\mathbf{x})}{D_{max}(\mathbf{x})}$$

as the dimension N increases. For each K, let r denote the average value of the ratios $r(\mathbf{x})$,

$$r = \frac{\sum_{\mathbf{x}\in\mathcal{K}} r(\mathbf{x})}{K}$$

We also find the standard deviation for each average,

$$\sigma = \sqrt{\frac{1}{K}\sum_{\mathbf{x}\in\mathcal{K}}(\mathbf{x} - r)}.$$

For each metric L^1, L^2, and L^∞, we fix K and plot the average value r as a function of the dimension N with standard deviation bars around each average. We use values $N = 1, 2, \ldots, 10$ and $N = 10, 20, \ldots, 100$.

First, consider the city-block metric L^1.

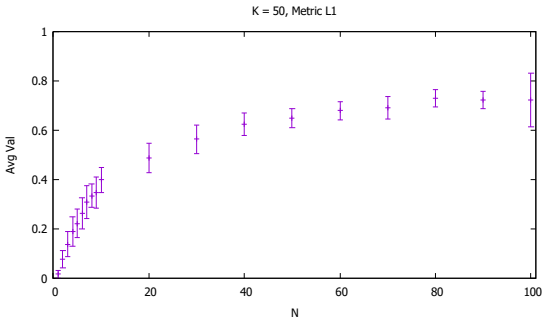

FIGURE 1. $K = 50$

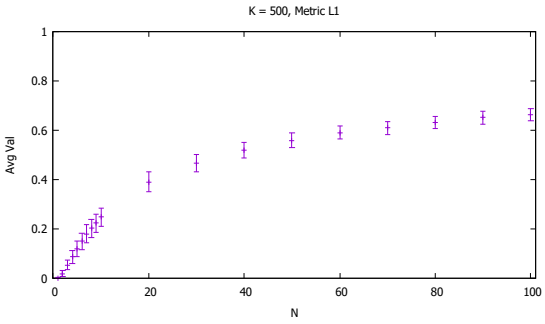

FIGURE 2. $K = 500$

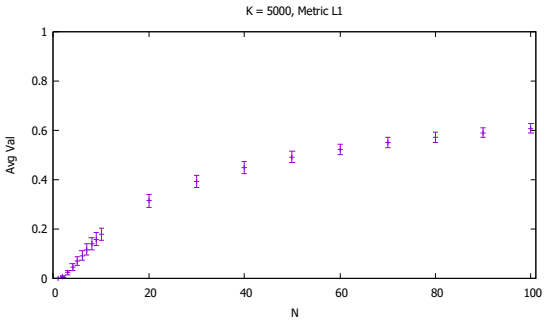

FIGURE 3. $K = 5000$

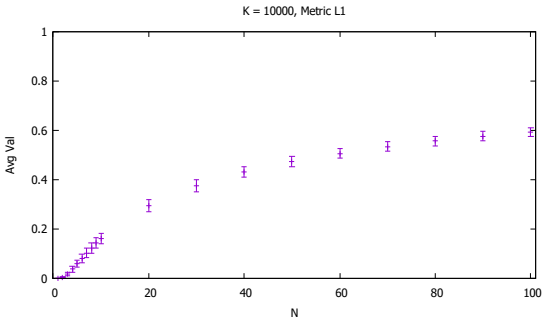

FIGURE 4. $K = 10000$

See Figures 49, 53, 55, and 56 in Appendix A for larger versions of these graphs, along with graphs for additional values of K. Here we see that while the ratio r does tend towards 1, it increases more slowly when we have a larger dataset. However, performance of this metric is not ideal, and already with the L^1 metric we see the effects of the curse of dimensionality.

Next, consider the Euclidean metric L^2:

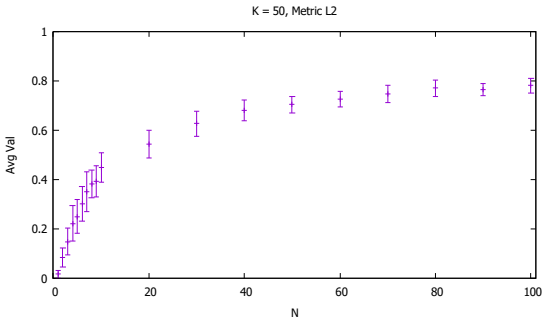

FIGURE 5. $K = 50$

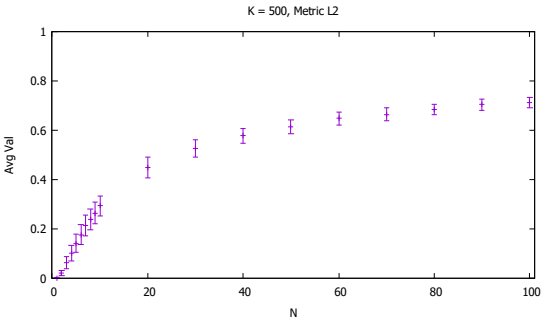

FIGURE 6. $K = 500$

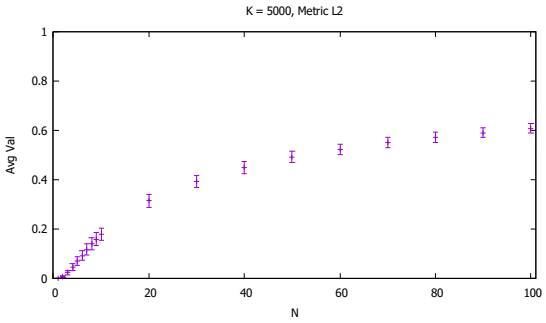

FIGURE 7. $K = 5000$

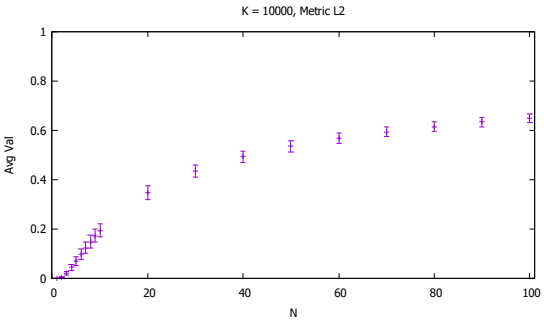

FIGURE 8. $K = 10000$

See Figures 57, 61, 63, and 64 in Appendix A for larger versions of these graphs and graphs for additional values of K. Again we see that the ratio r tends towards 1, but now we do not see as significant of a decrease in the speed with which this increases for larger datasets. These tests fall in line with the results in [1] that the L^1 metric performs better in high dimensional spaces than the L^2 metric. This runs counter to intuition we have to think of distance in terms of Euclidean metric in lower dimensions.

Next, let us consider the L^∞ metric:

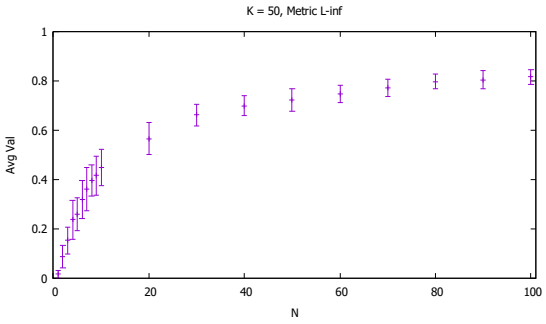

FIGURE 9. $K = 50$

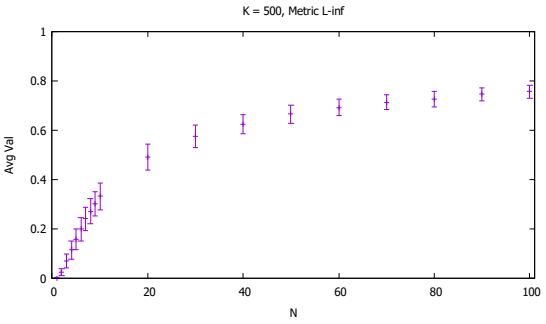

FIGURE 10. $K = 500$

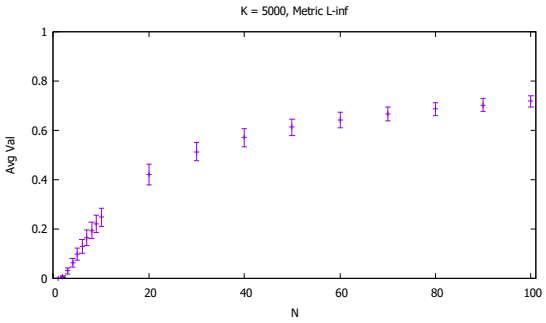

FIGURE 11. $K = 5000$

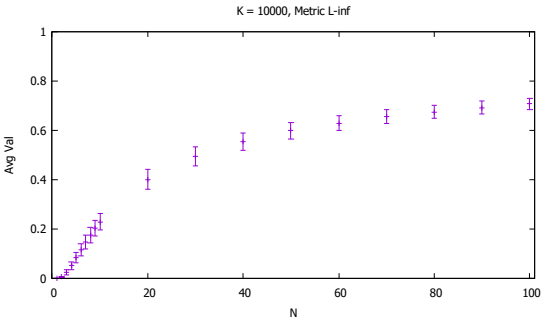

FIGURE 12. $K = 10000$

See Figures 65, 69, 71, and 72 in Appendix A for larger versions of these graphs, along with graphs for additional values of K. Here we do not see a significant increase in effectiveness for large values of K, and the ratio tends quickly towards infinity as the dimension N increases. These results further support the theoretical findings in [1], that L^2 is more effective at analyzing data in high dimensions than L^∞.

3.2. **Classification Accuracy.** Next we want to perform an experiment analyzing classification accuracy in high dimensional spaces. Let N be the dimension of the space. First, we generate a set $\mathcal{K}$ of K N-dimensional points from $U(0,1)^N$. For each $\mathbf{x} \in \mathcal{K}$, we randomly choose a class c_1 or c_2. Next, we generate a set $\mathcal{M}$ of M N-dimensional points from $U(0,1)^N$. For each $\mathbf{y} \in \mathcal{M}$, we find the closest point $\mathbf{x} \in \mathcal{K}$ to $\mathbf{y}$ using distance metric d and set the true class of $\mathbf{y}$ equal to the class of $\mathbf{x}$.

To test the classification accuracy, we now perturb each point $\mathbf{y} \in \mathcal{M}$ by ξ. More specifically, we let $\xi \in N(0, \sigma^2 I)$ and set $\tilde{\mathbf{y}} = \mathbf{y} + \xi$. Call this set of perturbed points $\tilde{\mathcal{M}}$. For each $\tilde{\mathbf{y}} \in \tilde{\mathcal{M}}$, set the assigned class of $\tilde{\mathbf{y}}$ equal to the class of its nearest neighbor in $\mathcal{K}$.

Let $P_{TA}(c_1, c_2)$ denote the probability that each $\tilde{\mathbf{y}}$ has assigned class c_2 and the associated $\mathbf{y}$ true class c_1. Using this we generate the confusion matrix:

		Assigned Class	
		c_1	c_2
True Class	c_1	$P_{TA}(c_1, c_1)$	$P_{TA}(c_2, c_1)$
	c_2	$P_{TA}(c_1, c_2)$	$P_{TA}(c_2, c_2)$

From this, generate from this graphs displaying the probability of correct identification, the accuracy rate P_C,

$$P_C = P_{TA}(c_1, c_1) + P_{TA}(c_2, c_2),$$

for varying values of N, with fixed values of K, M, and σ. We run the test with values $\sigma = 0.1$ and $\sigma = 0.5$, for each metric L^1, L^2, and L^∞.

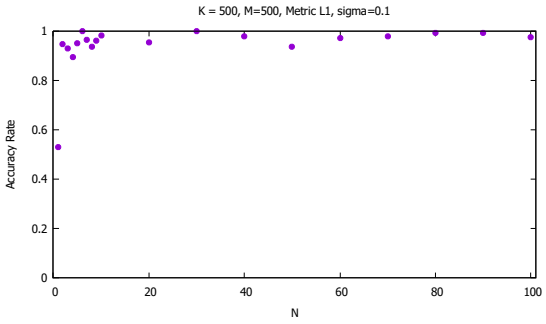

FIGURE 13. $K = 500$, $M = 500, \sigma = 0.1$

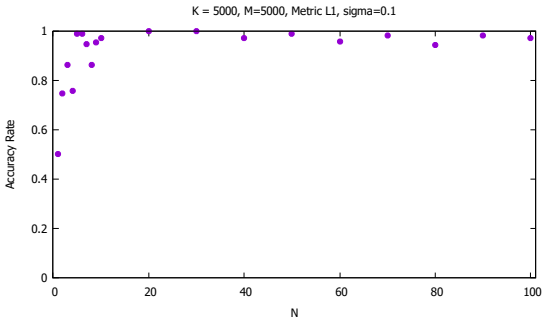

FIGURE 14. $K = 5000$, $M = 5000$, $\sigma = 0.1$

See Figures 73, 76 in Appendix B for larger versions of these graphs and additional graphs.

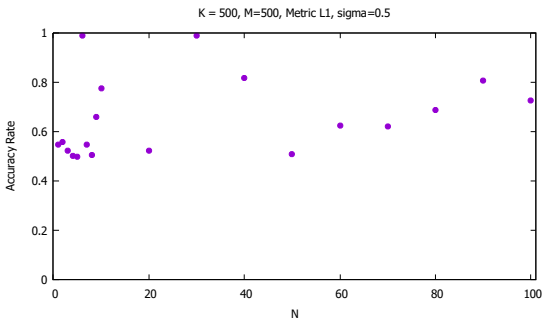

FIGURE 15. $K = 500$, $M = 500, \sigma = 0.5$

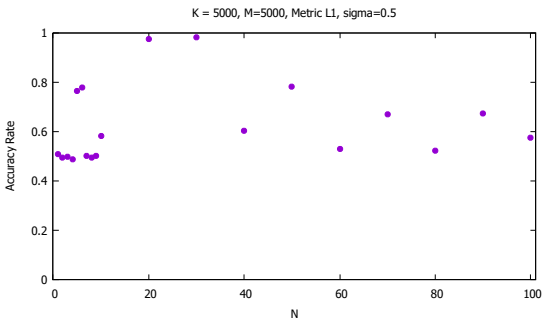

FIGURE 16. $K = 5000$, $M = 1000$, $\sigma = 0.5$

We see that for very small σ, such as $\sigma = 0.1$, the classification accuracy remains relatively high in high dimensions, while giving low values for low dimensions. However, as σ increases, the classification accuracy becomes wildly inaccurate.

Next, we repeat the experiment in the L^2 metric.

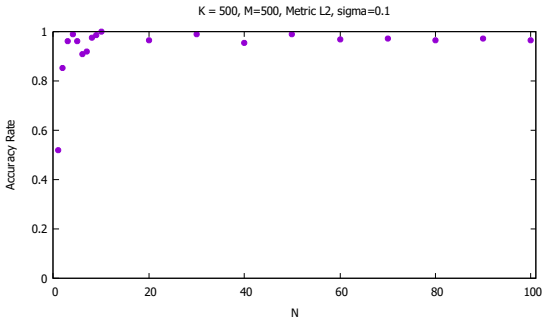

FIGURE 17. $K = 500$, $M = 500$, $\sigma = 0.1$

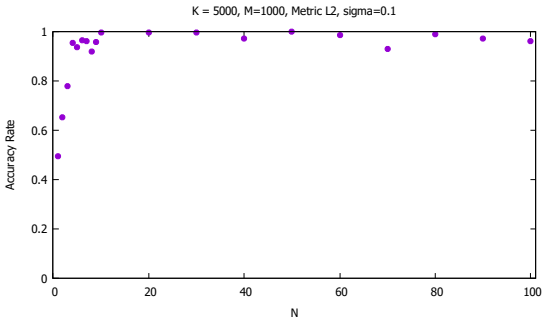

FIGURE 18. $K = 5000$, $M = 1000$, $\sigma = 0.1$

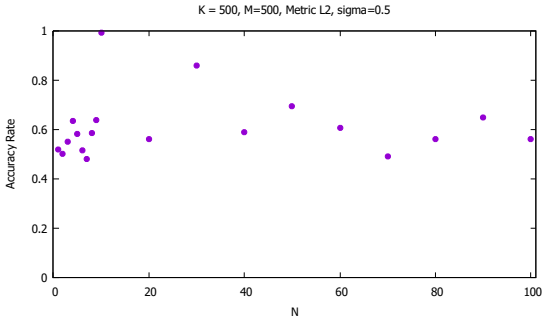

FIGURE 19. $K = 500$, $M = 500, \sigma = 0.5$

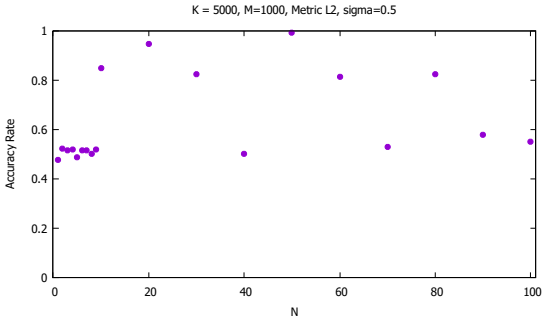

FIGURE 20. $K = 5000$, $M = 1000$, $\sigma = 0.5$

See Figures 81, 82, 83, 84 in Appendix B for larger versions of these graphs.

Last, we repeat the experiment with the L^∞ metric:

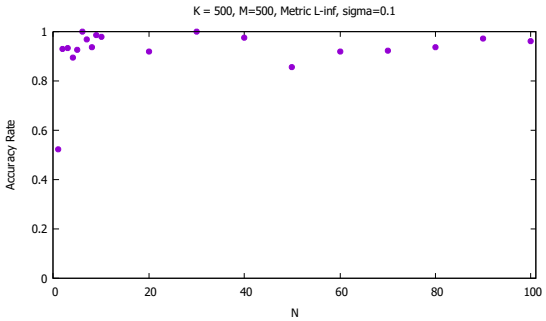

FIGURE 21. $K = 500$, $M = 500$, $\sigma = 0.1$

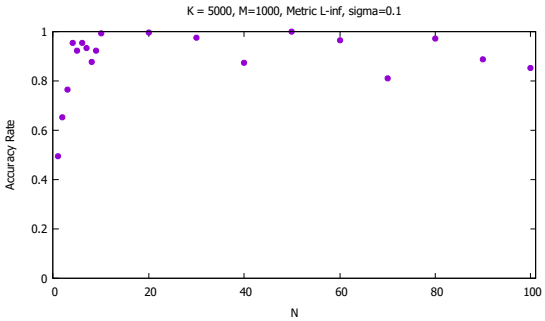

FIGURE 22. $K = 5000$, $M = 1000$, $\sigma = 0.1$

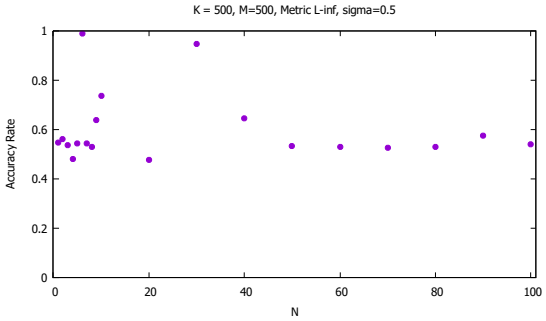

FIGURE 23. $K = 500$, $M = 500, \sigma = 0.5$

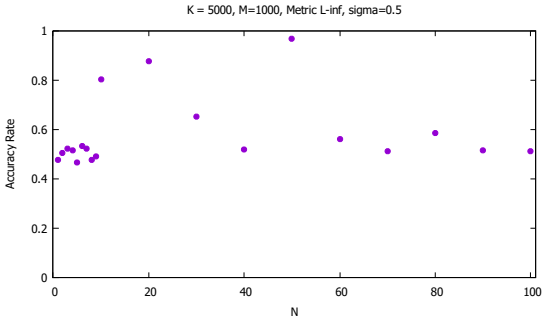

FIGURE 24. $K = 5000$, $M = 1000$, $\sigma = 0.5$

See Figures 86, 87, 90, 91 in Appendix B for larger versions of these graphs. We see the same behavior for the L^2 and L^∞ metrics that we saw for the L^1 metric: namely, that for small σ, the accuracy remains relatively high. However, the larger the value of σ, the less accurate our classification for perturbed data becomes. The value taken for K appears to have little effect on this outcome.

Lastly, consider the following:

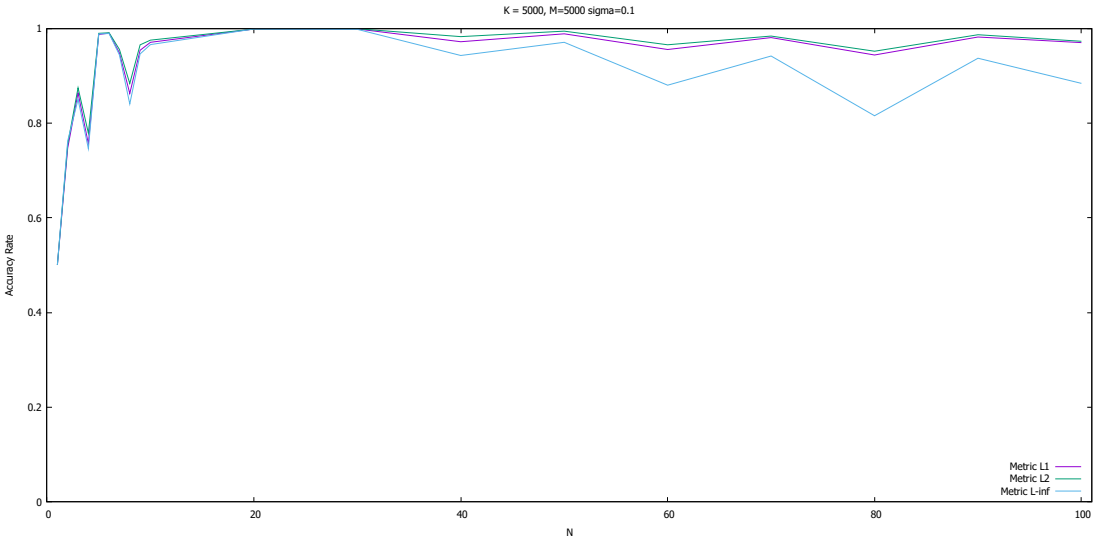

FIGURE 25. $K = 5000$, $M = 5000$, $\sigma = 0.1$

The above graph shows the classification accuracy for L^1, L^2, and L^∞. Observe that the classification accuracy for L^∞ is lower than for the other two metrics. One possibility for this is that the L^∞ norm measures the distance between two points on only one component. It is therefore easier for the small peturbation on our datapoints to result in the points having a different nearest neighbor.

3.3. **Maximum and Minimum Distance With Gaussian Distribution.** Lastly, we would like to repeat the first experiment, but this time draw our random points from Gaussian$(0, 1)$ instead of the standard uniform distribution on the interval from 0 to 1. In this experiment, it is important that we normalize our data to norm 1 to prepare it for modeling.

Let N be the dimension of the space and K the number of points sampled. We generate a set $\mathcal{K}$ of K N-dimensional points in Gaussian$(0, 1)$. Next, we normalize each point to norm 1. To do this let $\mathbf{x}_i = (x_1, x_2, \ldots, x_K)$ for $1 \leq i \leq K$. Let $\min_i = \min_{1 \leq i \leq K}\{x_i\}$ and $\max_i = \max_{1 \leq i \leq K}\{x_i\}$.

We normalize our datapoints in two ways: first, we divide each component of the vector by the vector's Euclidean length. Then, we repeat the experiment with max-min normalization, where we normalize each point by changing the value of x_i as follows:

$$x_i = \frac{x_i - \min_i}{\max_i - \min_i}.$$

Next, we find the distance to the nearest neighbor and furthest neighbor as in the first maximum and minimum distance experiment to find the ratio $r = D_{min}/D_{max}$ along with the standard deviation σ. We plot the average value of r as a function of N for $N = 1, 2, \ldots, 10$ and $N = 10, 20, \ldots, 100$ for metrics L^1, L^2, and L^∞.

First, we plot the graphs where we normalized with respect to the vectors' Euclidean length.

With the city block metric L^1 and Euclidean normalization we obtain:

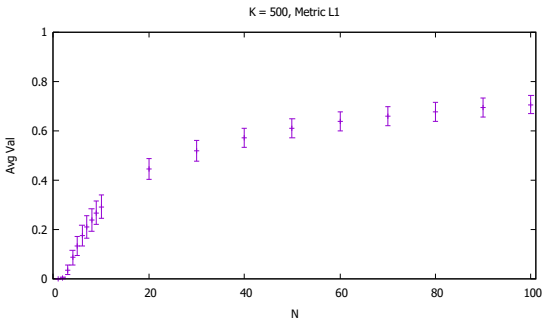

FIGURE 26. $K = 50$

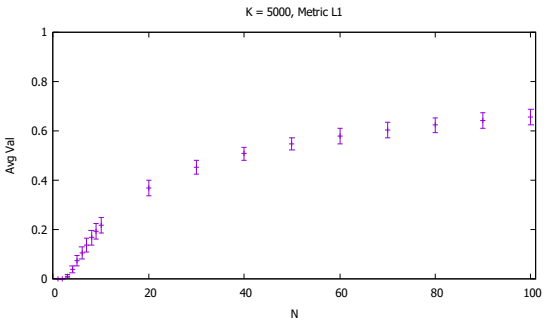

FIGURE 27. $K = 5000$

See Figures 109 and 110 in Appendix D for larger versions of these graphs. We observe that as expected from [1], for large datasets the behavior of the nearest neighbor to the Gaussian distribution behaves similarly to results for data points from the uniform distribution for metric L^1.

For metric L^2 and Euclidean normalization we obtain the following results:

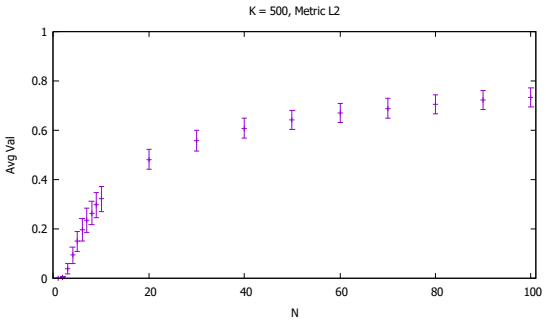

FIGURE 28. $K = 500$

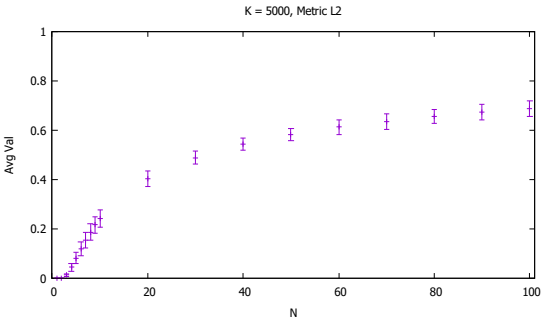

FIGURE 29. $K = 5000$

See Figures 111 and 112 in Appendix D for larger versions of these graphs. Again, the ratio behaves similarly to that of points sampled from the uniform distribution using metric L^2.

Lastly, for the maximum distance metric L^∞ we obtain

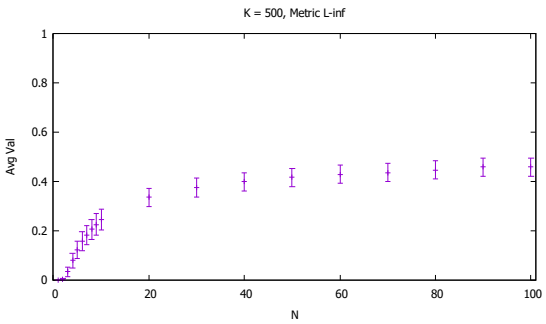

FIGURE 30. $K = 500$

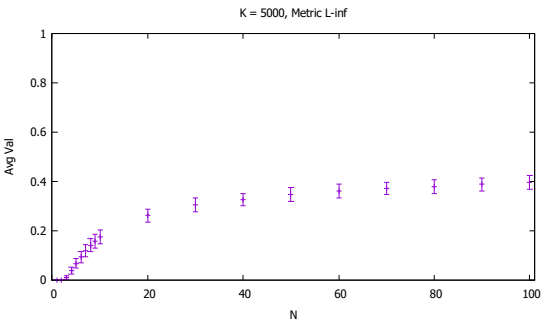

FIGURE 31. $K = 5000$

See Figures 113 and 114 in Appendix D for larger versions of these graphs. We see that the L^∞ metric provides better results than the L^1 and L^2 metrics under the Gaussian distribution.

Next, we proceed to look at the ratios when we applied max-min normalization. With the city block metric L^1 and max-min normalization we obtain:

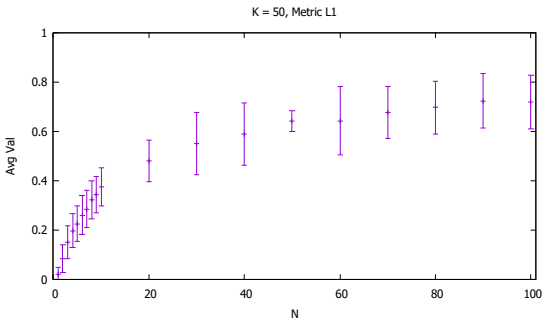

FIGURE 32. $K = 50$

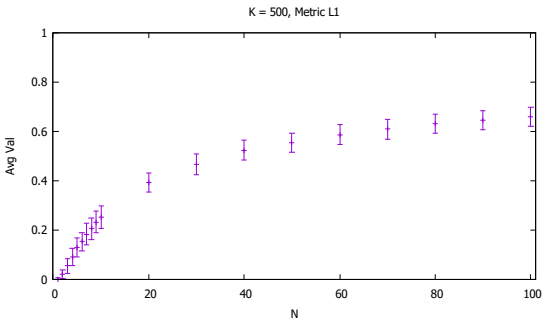

FIGURE 33. $K = 500$

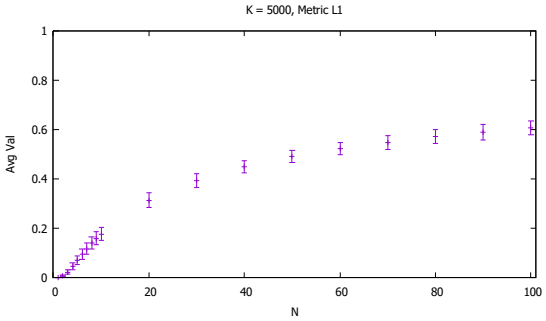

FIGURE 34. $K = 5000$

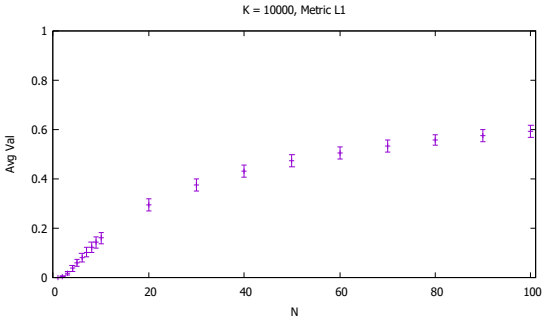

FIGURE 35. $K = 10000$

See Figures 92, 94, 96, and 97 in Appendix C for larger versions of these graphs and graphs for additional values of K. We observe that for small amounts of data points, our analysis yields an incredibly high standard deviation. However, as expected from [1], for large datasets the behavior of the nearest neighbor to the Gaussian distribution behaves similarly to results for data points from the uniform distribution for metric L^1.

For metric L^2 we obtain the following results:

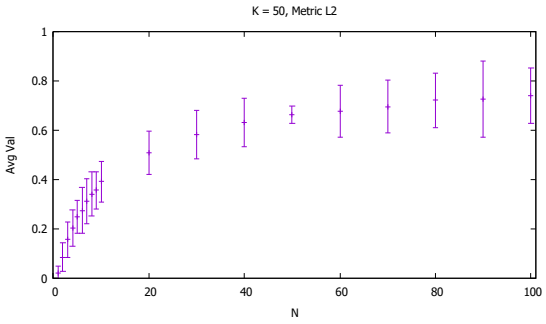

FIGURE 36. $K = 50$

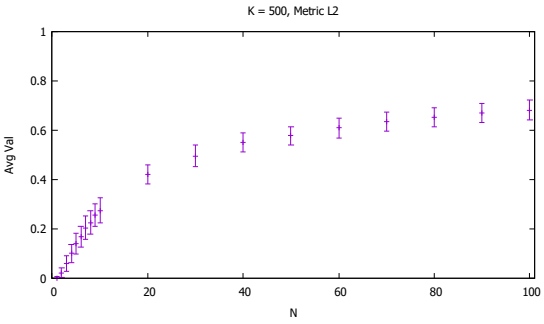

FIGURE 37. $K = 500$

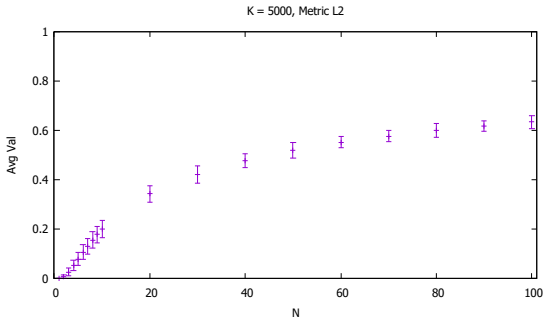

FIGURE 38. $K = 5000$

See Figures 98, 100, and 102 in Appendix C for larger versions of these graphs. Again, for small sets of data points our analysis yields a high standard deviation, but for larger sets of data points the ratio behaves similarly to that of points sampled from the uniform distribution using metric L^2.

Lastly, for the maximum distance metric L^∞ we obtain

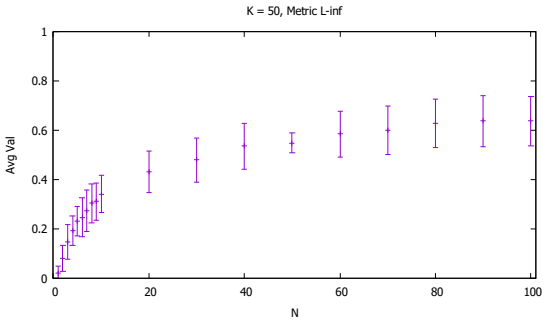

FIGURE 39. $K = 50$

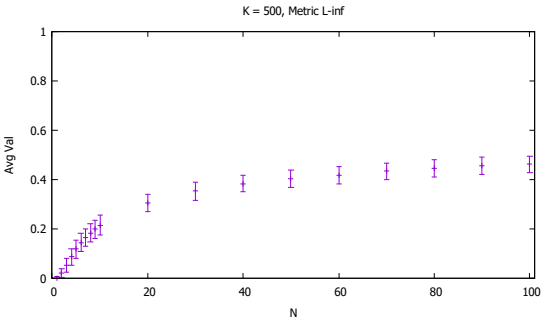

FIGURE 40. $K = 500$

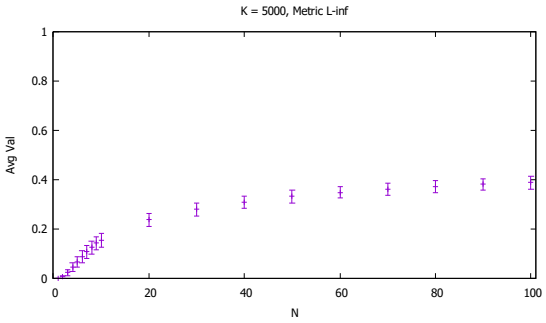

FIGURE 41. $K = 5000$

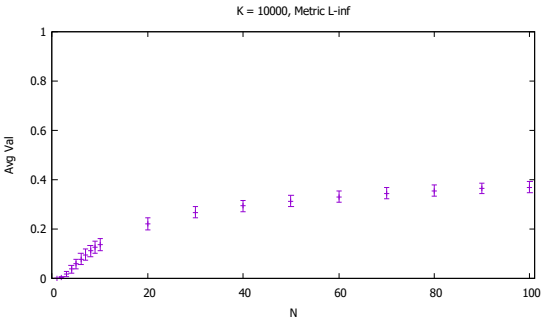

FIGURE 42. $K = 10000$

See Figures 103, 105, 107, and 108 in Appendix C for larger versions of these graphs. With this metric we again have a high standard distribution for smaller datasets. However, for large datasets, we see that the L^∞ metric provides better results than the L^1 and L^2 metrics under the Gaussian distribution. However, at dimension 100, the ratio is already near 0.4, suggesting that further research into more effective methods is still necessary.

Lastly, we compare the results between normalization with respect to Euclidean length and max-min normalization. First, observe the results for $K = 500$.

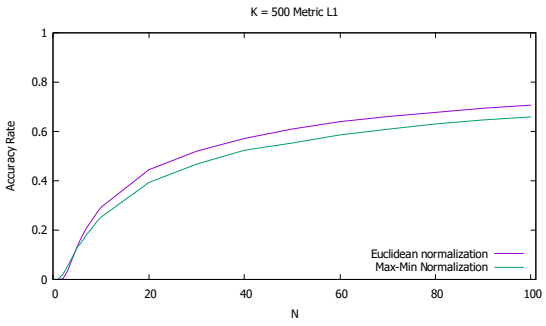

FIGURE 43. $K = 500$, Metric L^1

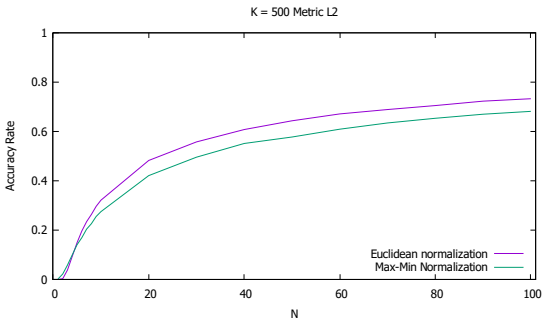

FIGURE 44. $K = 500$, Metric L^2

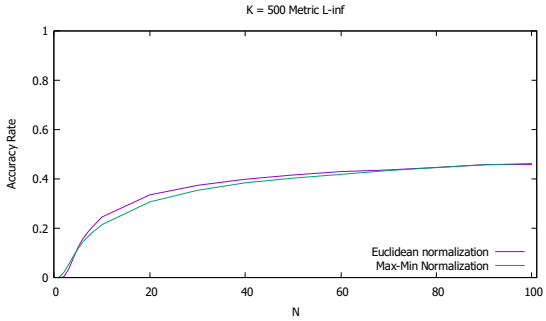

FIGURE 45. $K = 500$, Metric L^∞

See Figures 115, 116 and 117 in Appendix E for larger versions of these graphs.

Next observe the results of the comparison for $K = 5000$:

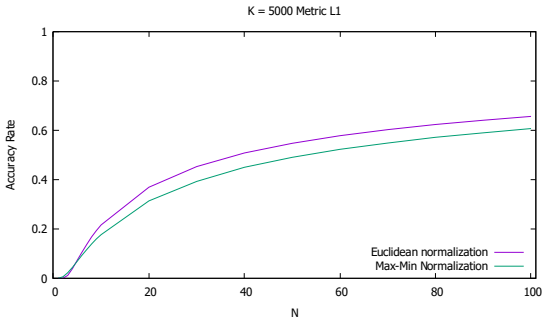

FIGURE 46. $K = 5000$, Metric L^1

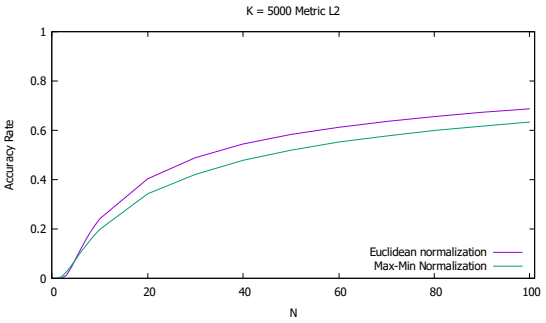

FIGURE 47. $K = 5000$, Metric L^2

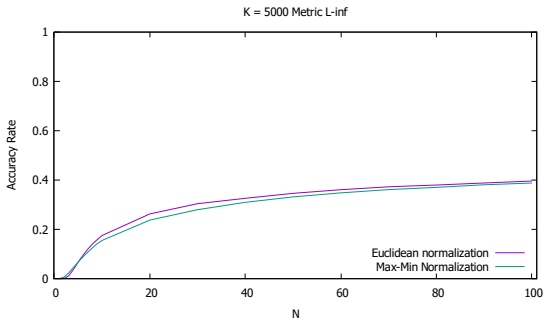

FIGURE 48. $K = 5000$, Metric L^∞

See Figures 118, 119 and 120 in Appendix E for larger versions of these graphs. It appears from our empirical results that normalization with respect to the Euclidean length of the vector yields higher accuracy than max-min normalization. However, with the L^∞ norm, this difference is less pronounced.

4. Conclusion

All in all, we see that the curse of dimensionality holds true for our empirical tests on theoretical results. In high dimensions, the ratio between the nearest neighbor and farthest neighbor grows closer to one as the dimension N increases for metrics L^1, L^2, and L^∞, whether our data points are taking from the uniform distribution or the Gaussian distribution. Furthermore, anything more than a very small perturbance in our data creates massively inaccurate classification of data, demonstrating that for analysis of high-dimensional data we need something more than the nearest-neighbor rule.

Ideas for theoretical results on which to conduct empirical tests on include local dimensionality reduction [2], fractional distance metrics [1], and k-nearest neighbor methods [8]. An especially promising route is the use of fractional distance metrics, with greatly increased performance of small fractional distance metrics reported in [1].

References

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim: On the Surprising Behavior of Distance Metrics in High Dimensional Spaces. *Database Theory ICDT 2001*, 2001.

[2] Charu C. Aggarwal and Chandan K. Reddy. *Data Clustering Algorithms and Applications*. CRC Press, 2014.

[3] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft: When is "Nearest Neighbor" Meaningful? *ICDT Conference Proceedings*, 1999.

[4] Richard Bellman. *Adaptive Control Processes*. Princeton University Press, 1961.

[5] Marcel Caraciolo: Data mining in practice: DataPreprocessing and the Use of Normalization. `http://aimotion.blogspot.com/2009/09/data-mining-in-practice.html`

[6] Robert M. Haralick: Discrete Bayes Pattern Recognition. Lecture Notes, Department of Computer Science, The Graduate Center, CUNY. `http://haralick.org/ML/discrete_bayes.pdf`

[7] Robert M. Haralick: High Dimensional Spaces. Lecture Notes, Department of Computer Science, The Graduate Center, CUNY. `http://haralick.org/ML/high_dimensional_spaces.pdf`

[8] Robert M. Haralick: Nearest Neighbor Rule. Lecture Notes, Department of Computer Science, The Graduate Center, CUNY. `http://haralick.org/ML/nearest_neighbor.pdf`

[9] John Hopcroft and Ravi Kannan. *Computer Science Theory*. Draft. `https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/hopcroft-kannan-feb2012.pdf`.

[10] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani: "Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces." *SIAM J. Comput.*, 2000.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Second Edition.* Springer, 2009.

[12] Panos M. Pardalos: Sparse Classification Methods for High Dimensional Data. Lecture notes, University of Florida. `http://www.hse.ru/data/2013/02/08/1307870298/HDDTalk.pdf`

Appendix A. Graphs For Project 1, Part 1

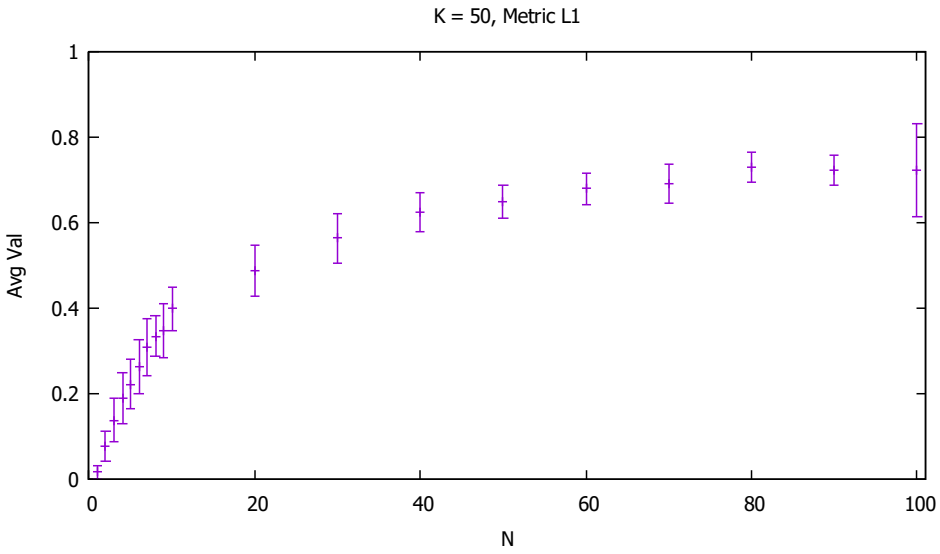

Figure 49

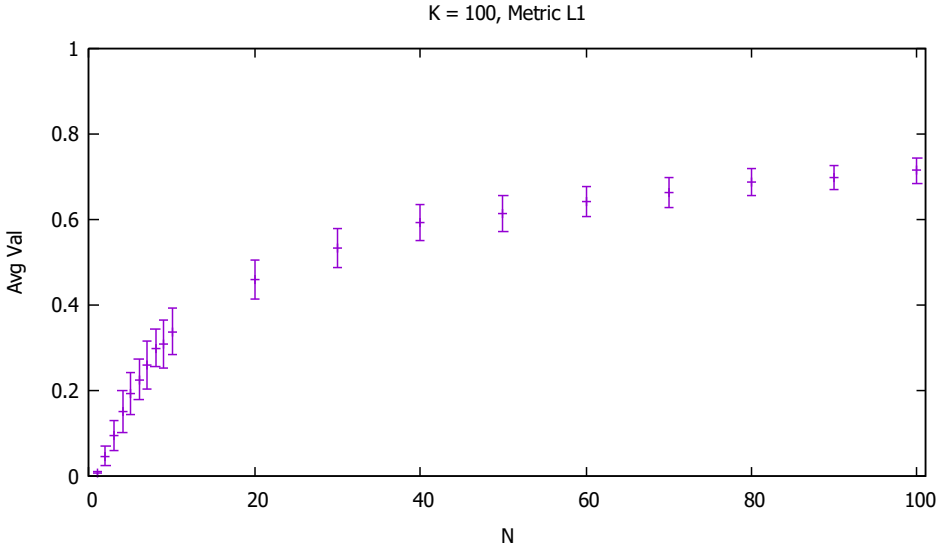

Figure 50

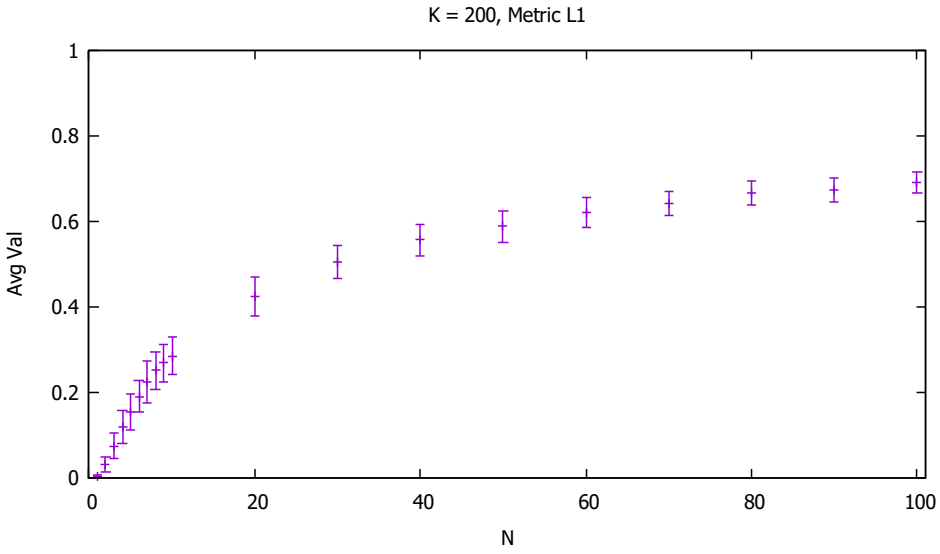

FIGURE 51

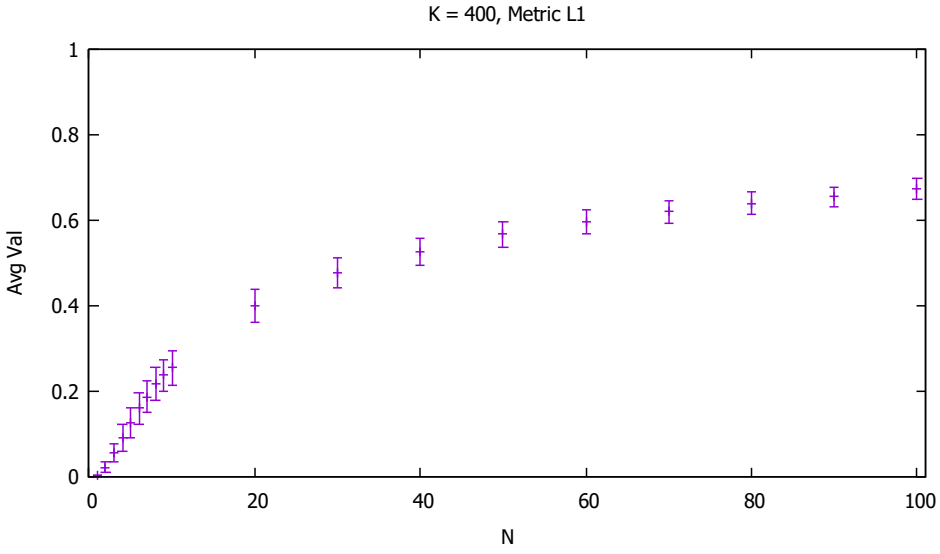

FIGURE 52

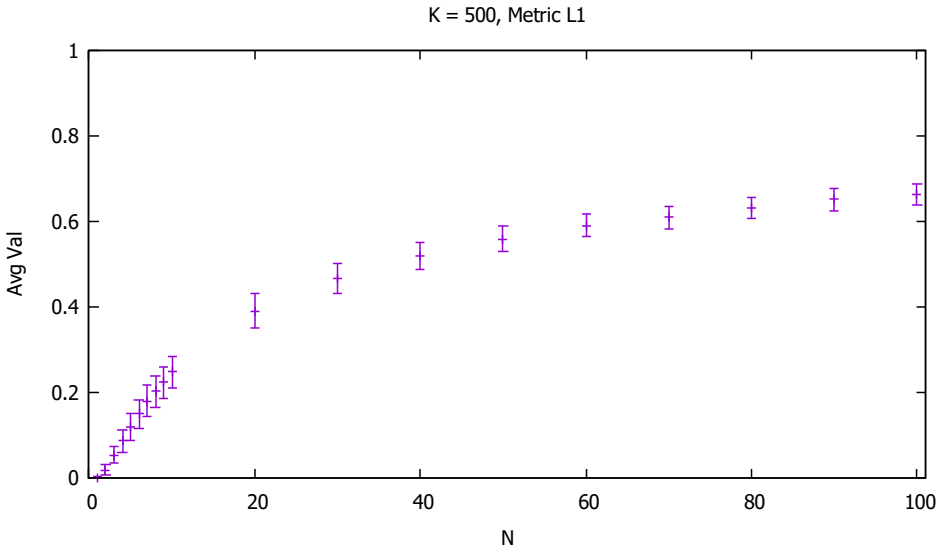

FIGURE 53

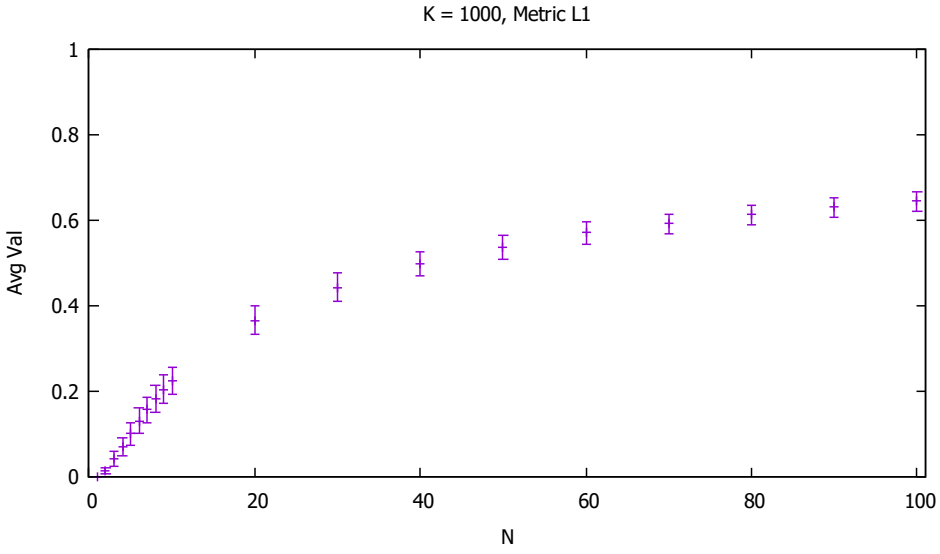

FIGURE 54

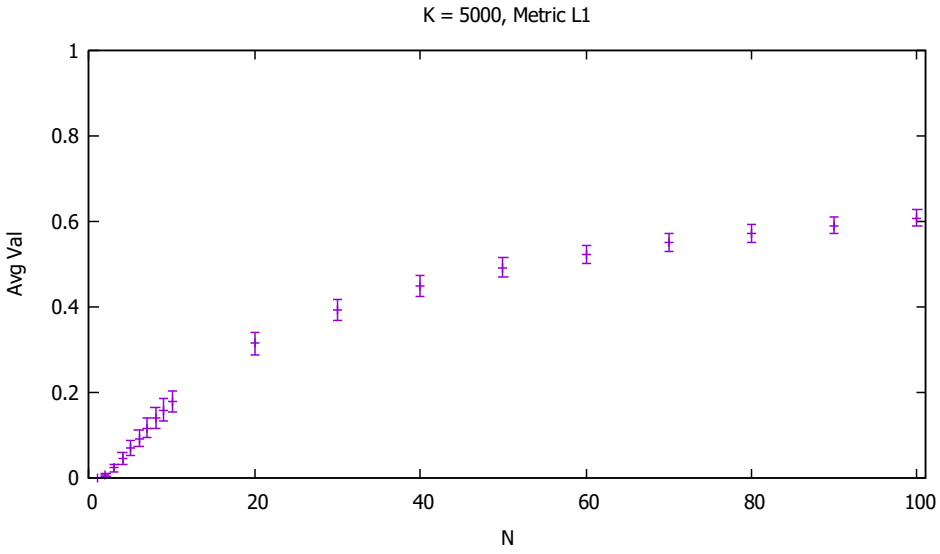

FIGURE 55

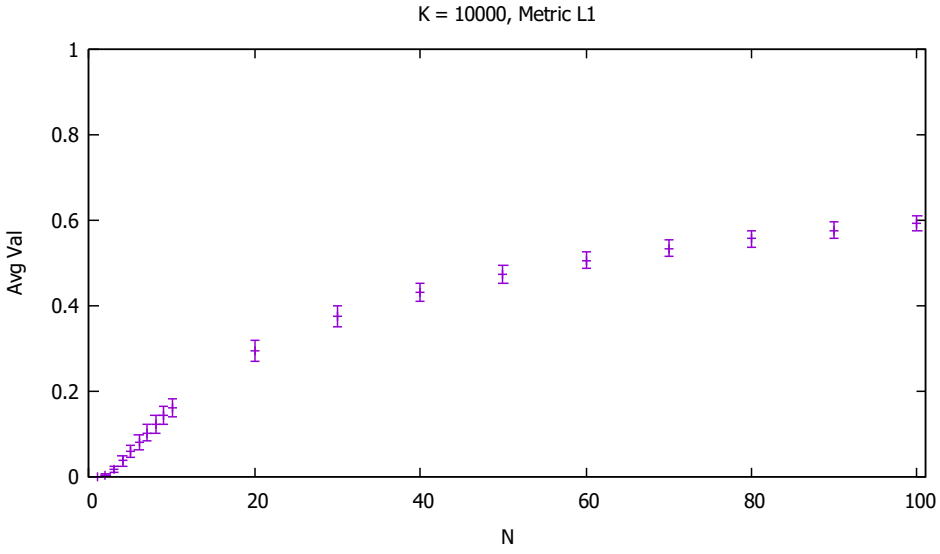

FIGURE 56

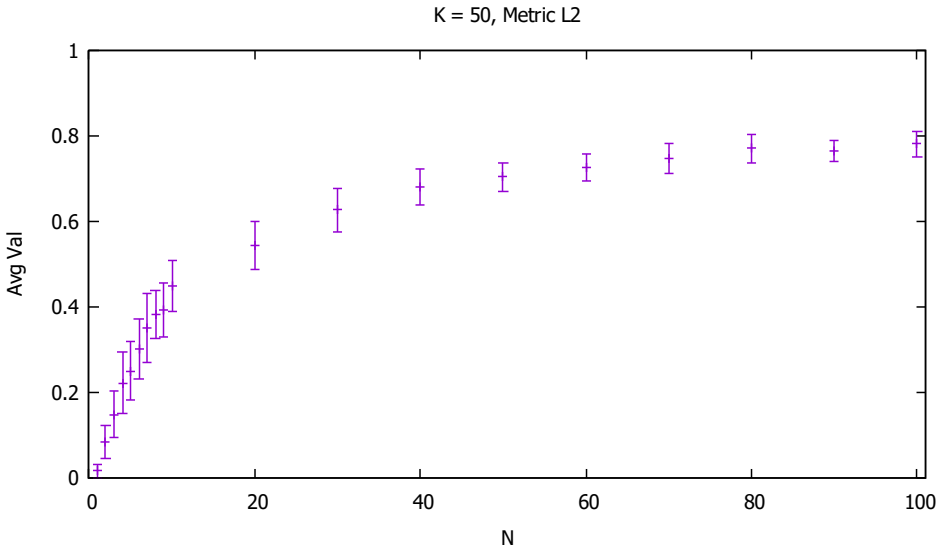

FIGURE 57

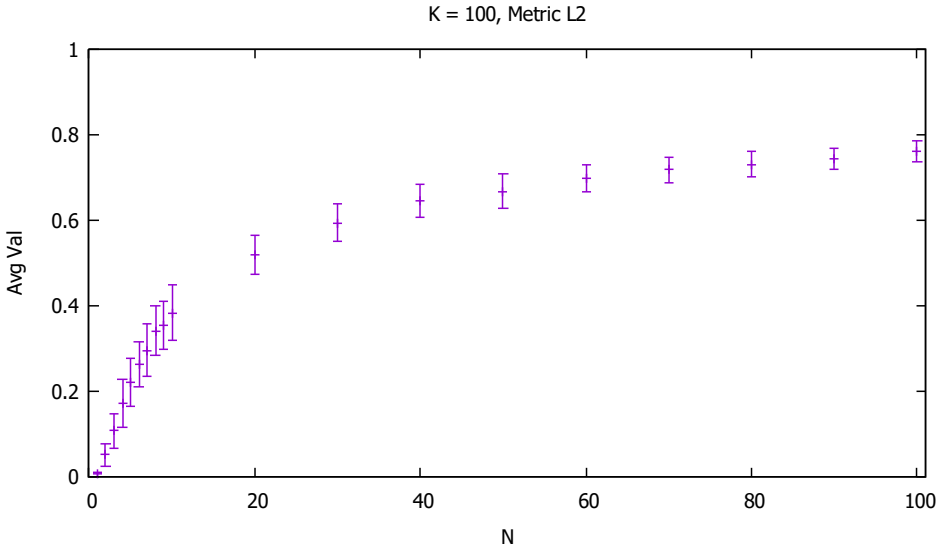

FIGURE 58

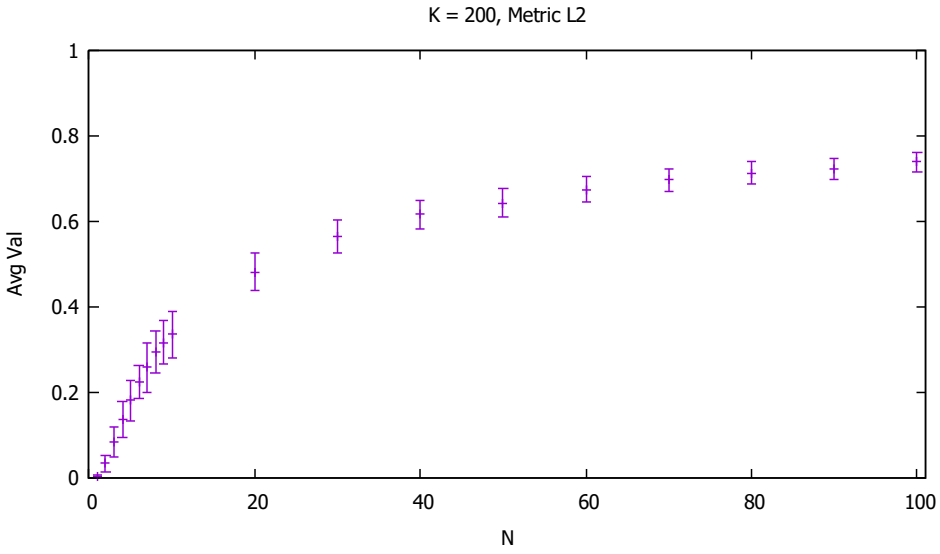

FIGURE 59

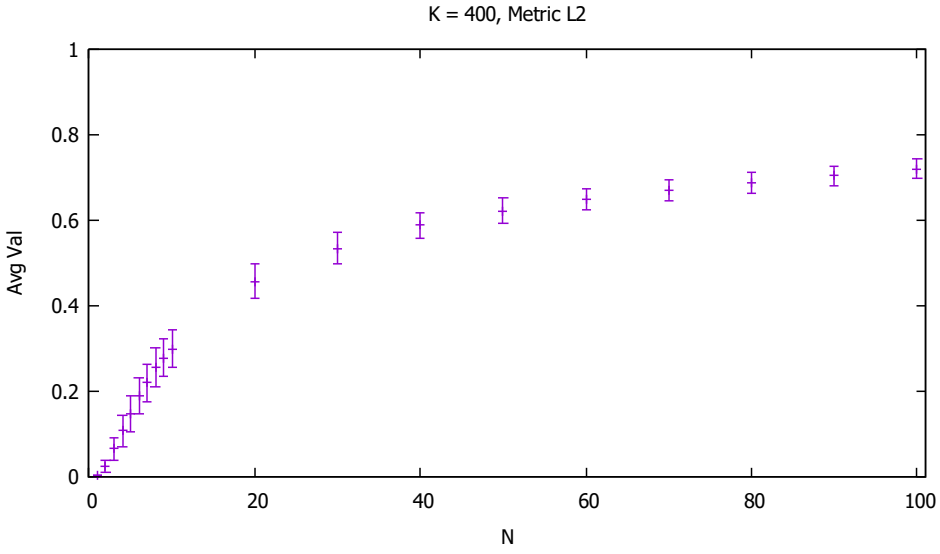

FIGURE 60

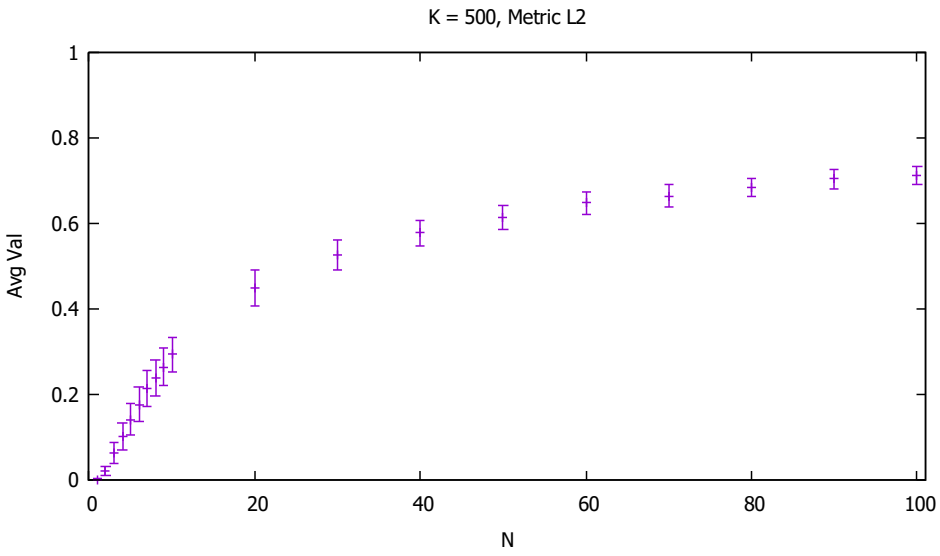

FIGURE 61

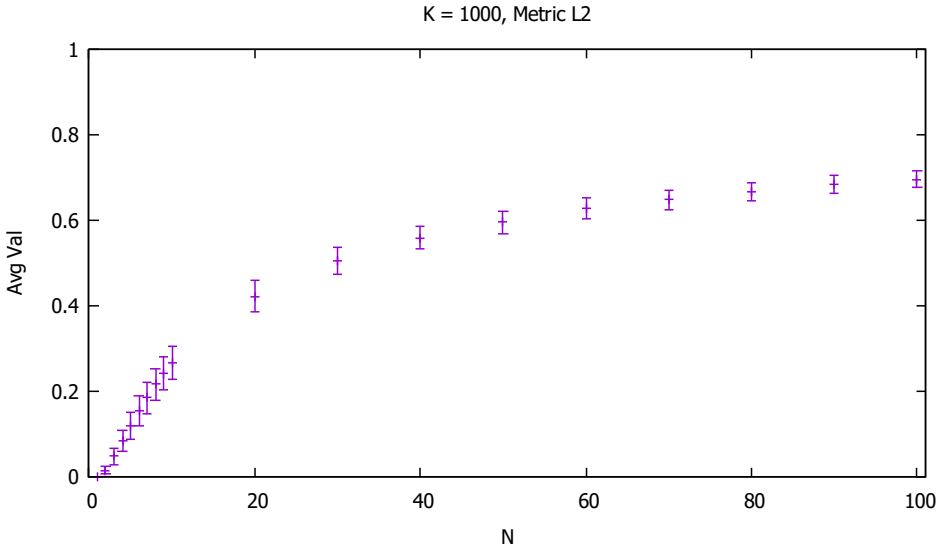

FIGURE 62

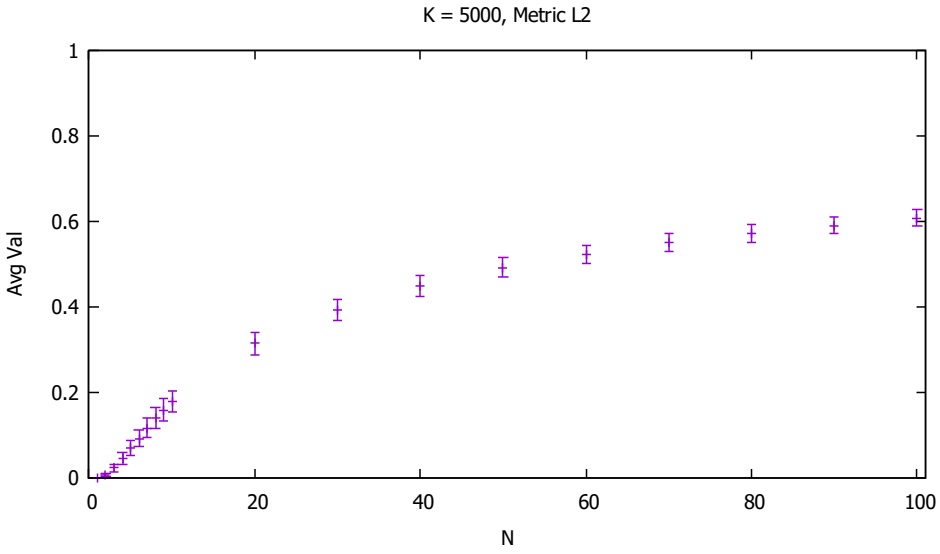

FIGURE 63

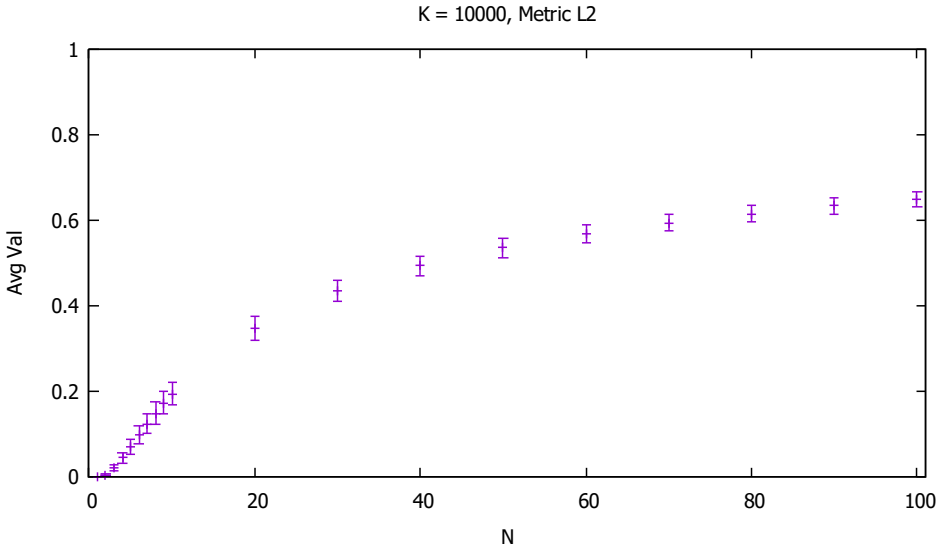

FIGURE 64

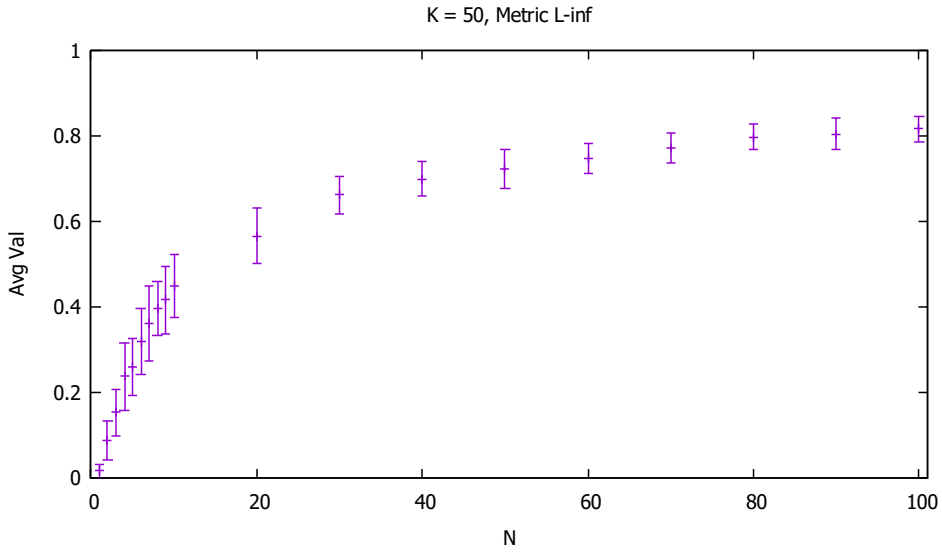

FIGURE 65

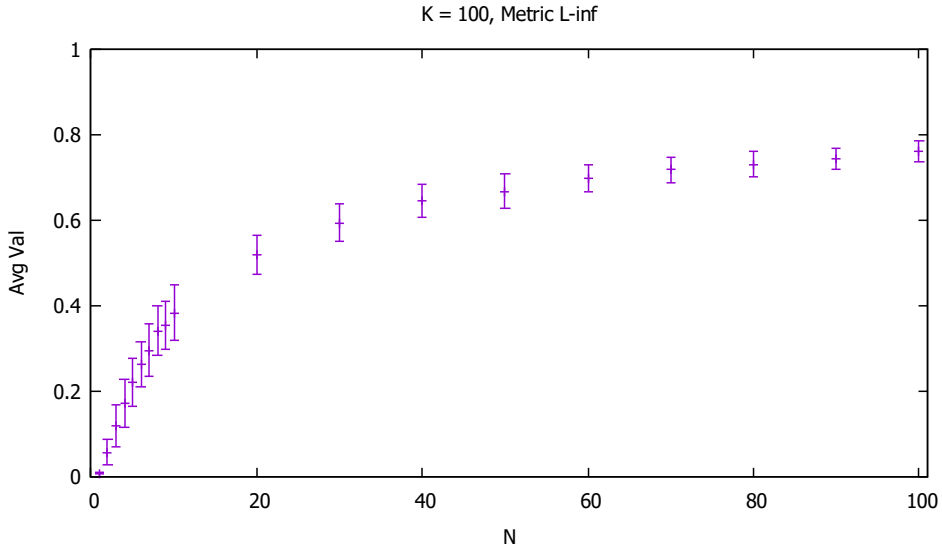

FIGURE 66

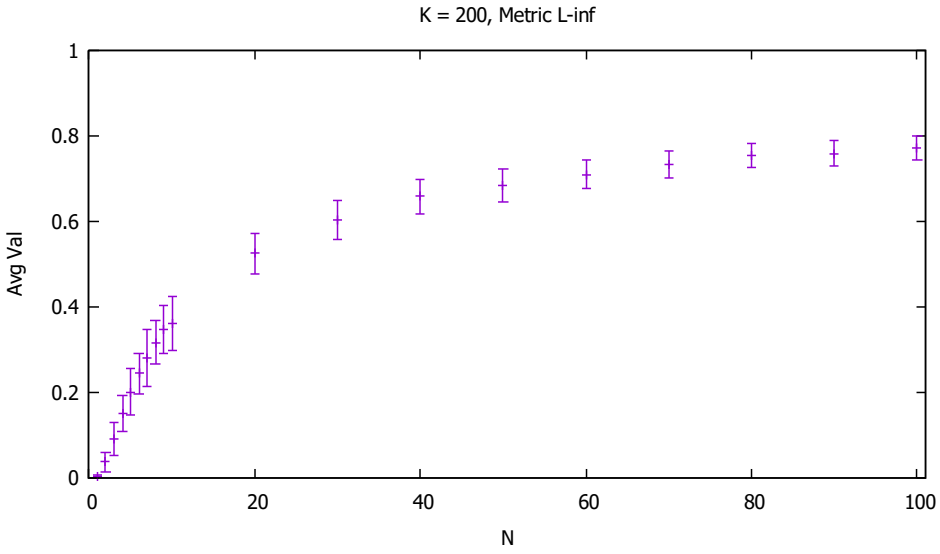

FIGURE 67

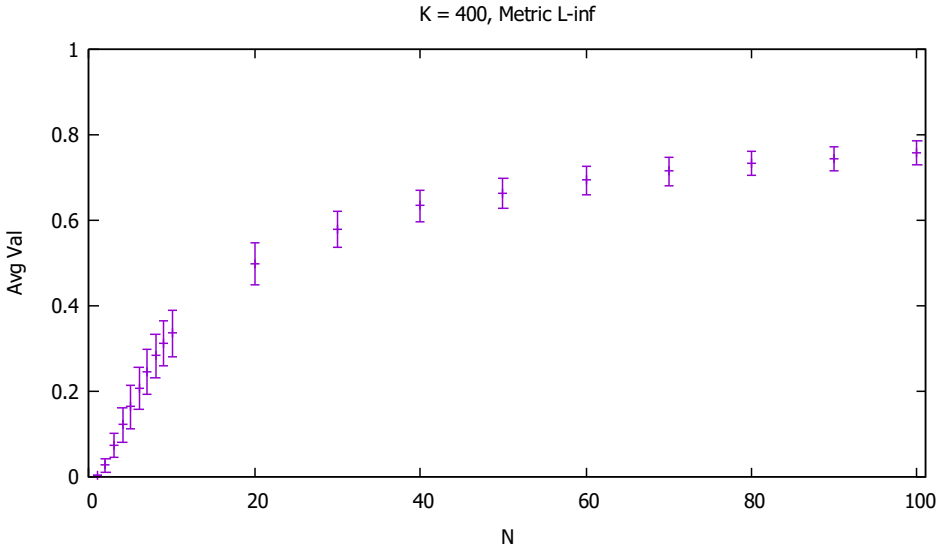

FIGURE 68

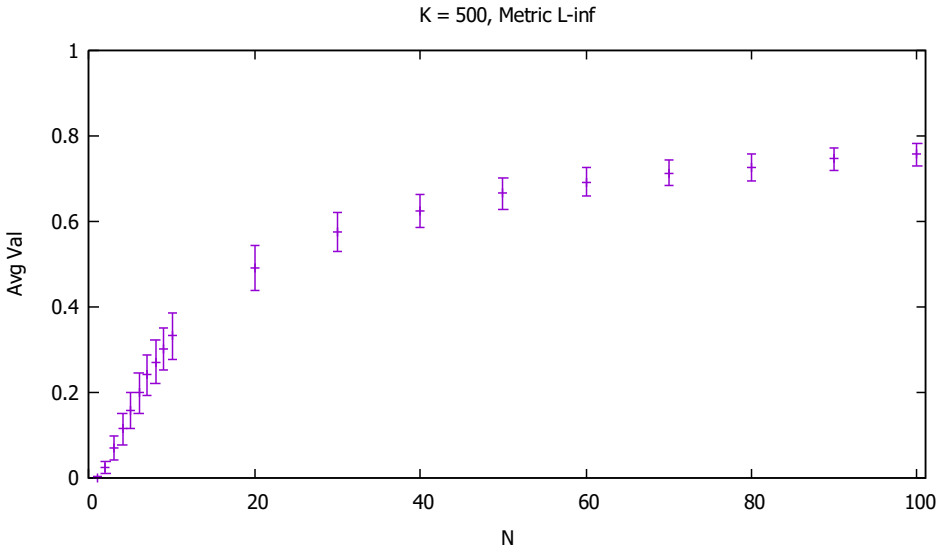

FIGURE 69

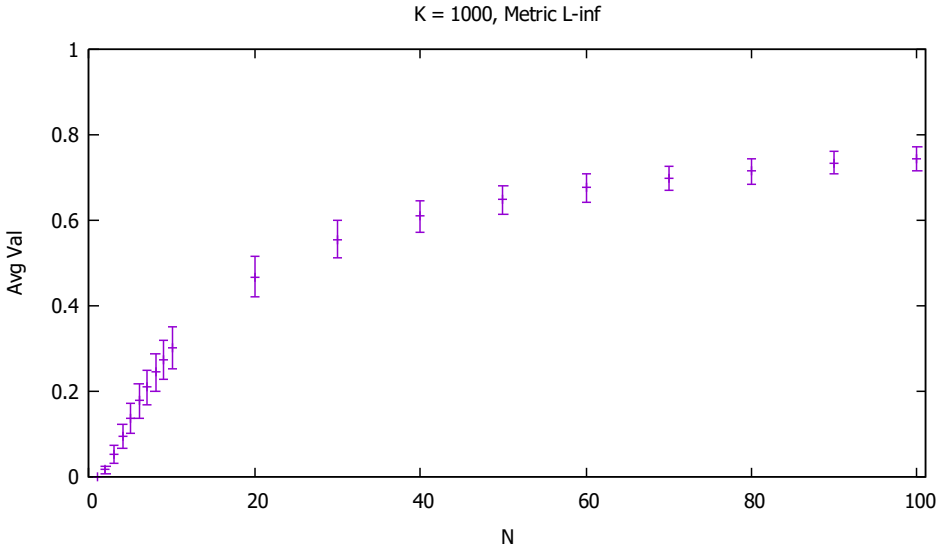

FIGURE 70

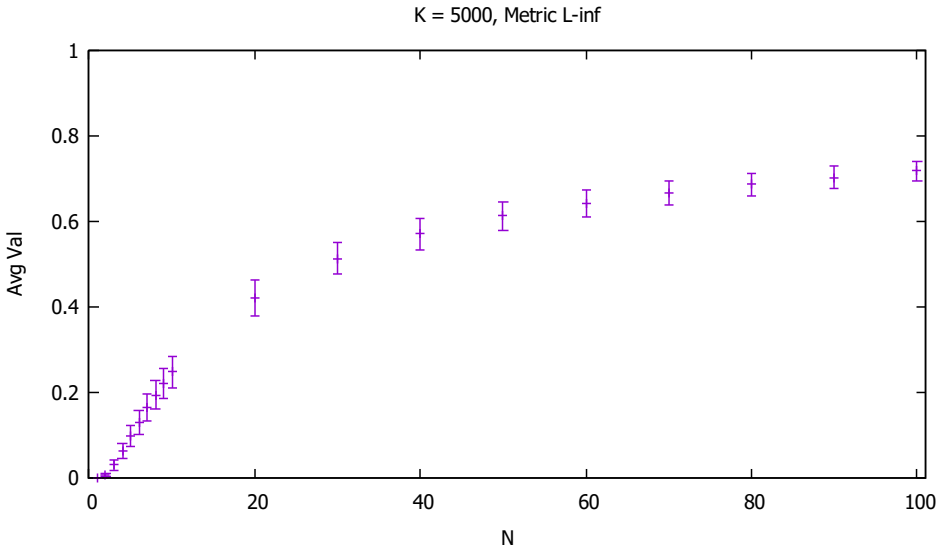

FIGURE 71

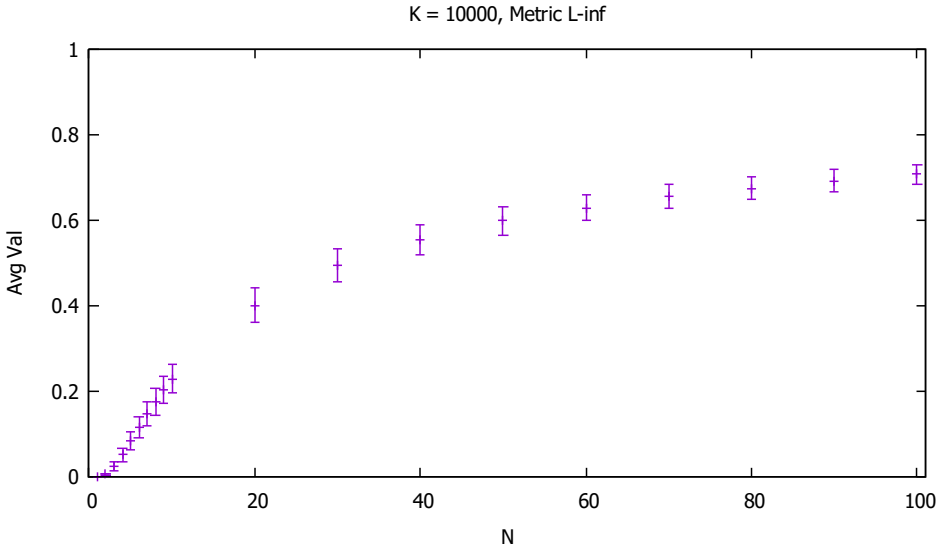

FIGURE 72

Appendix B. Graphs For Project 1, Part 2

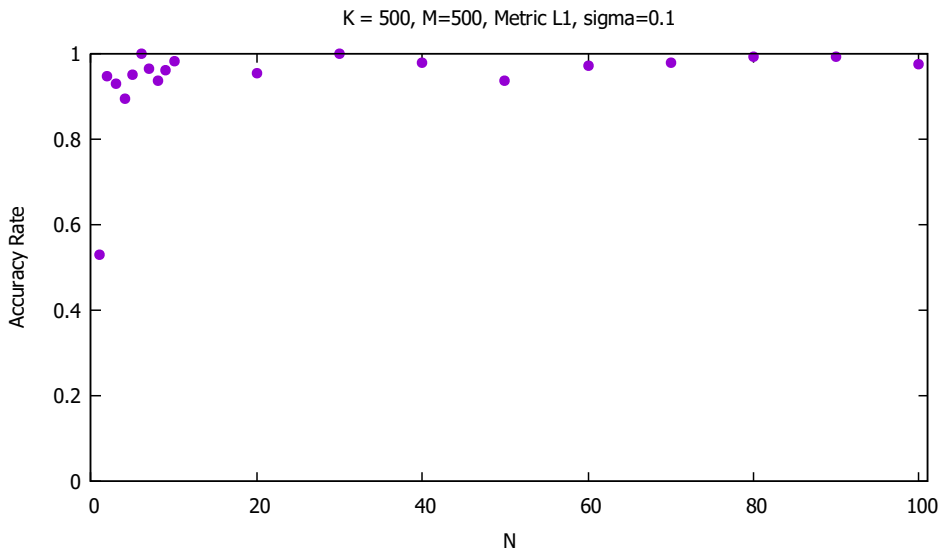

Figure 73

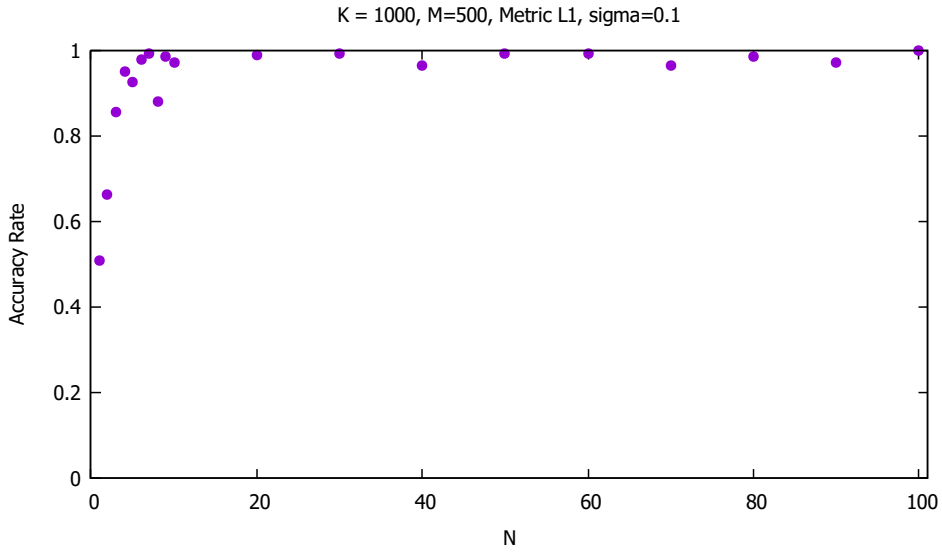

Figure 74

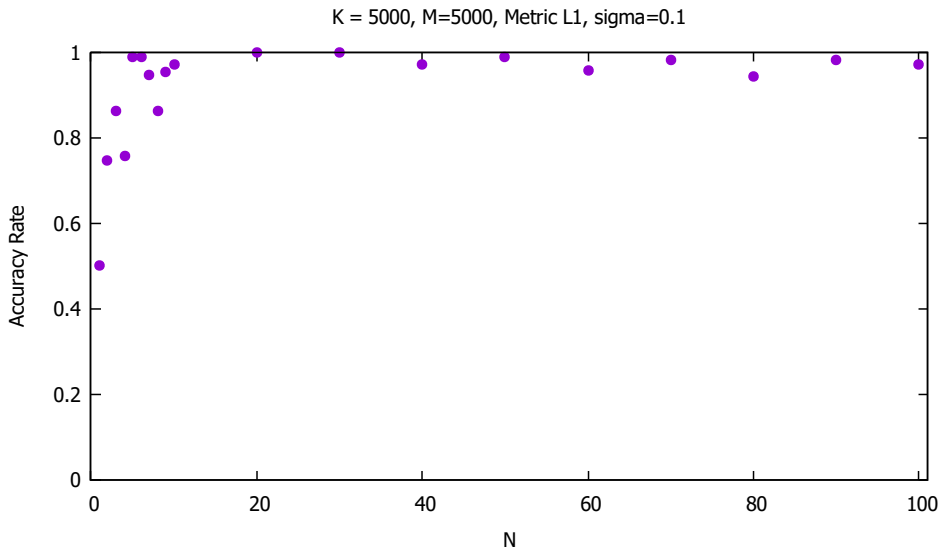

FIGURE 75

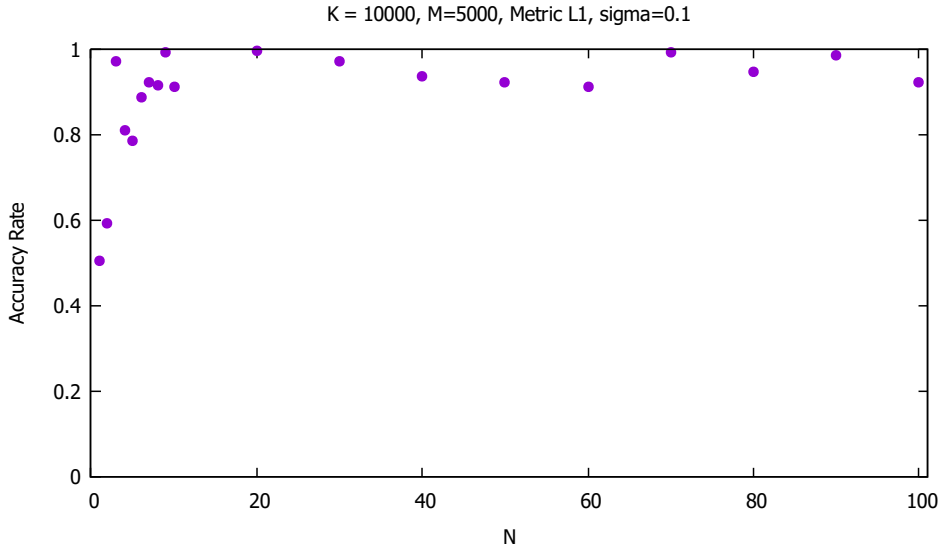

FIGURE 76

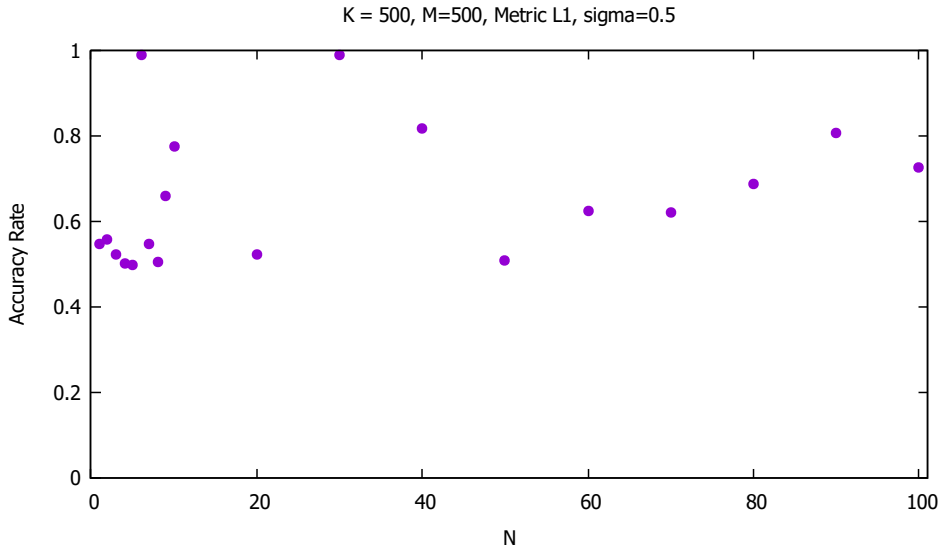

FIGURE 77

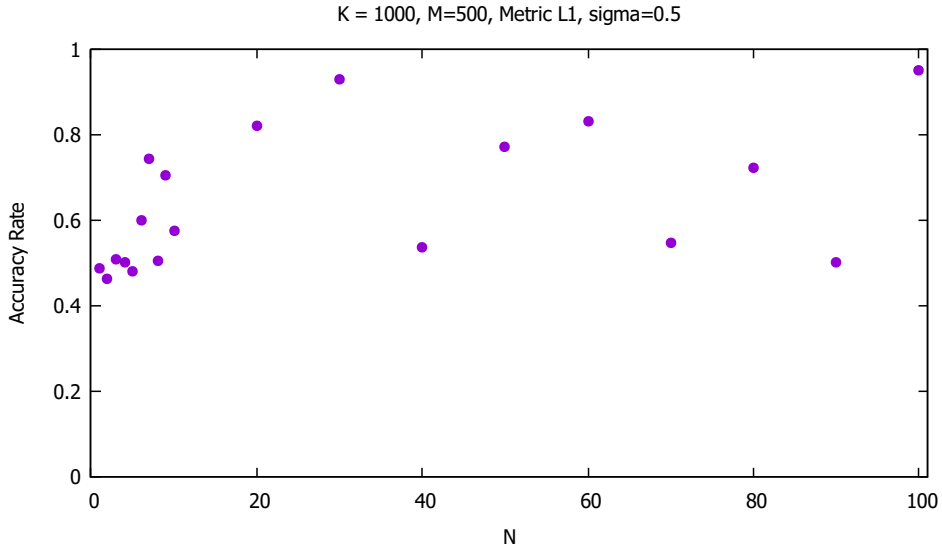

FIGURE 78

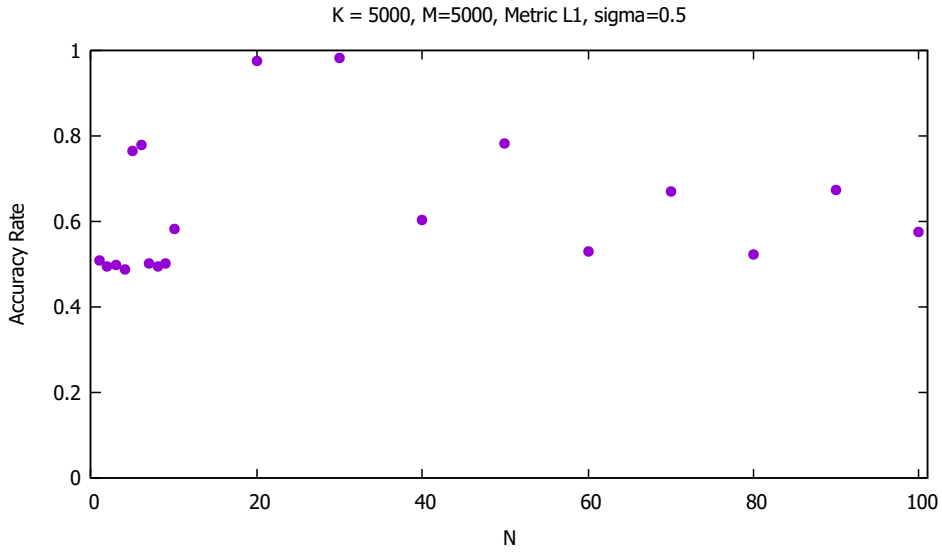

FIGURE 79

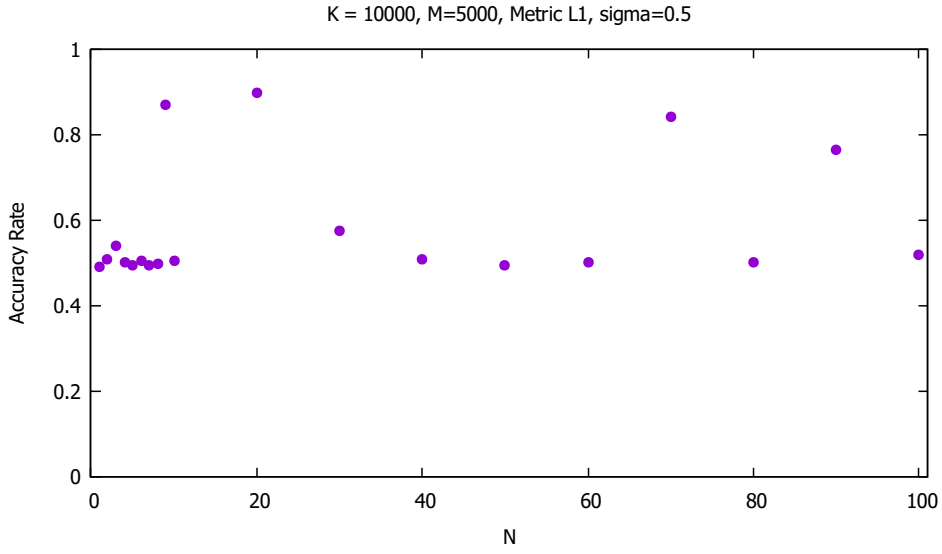

FIGURE 80

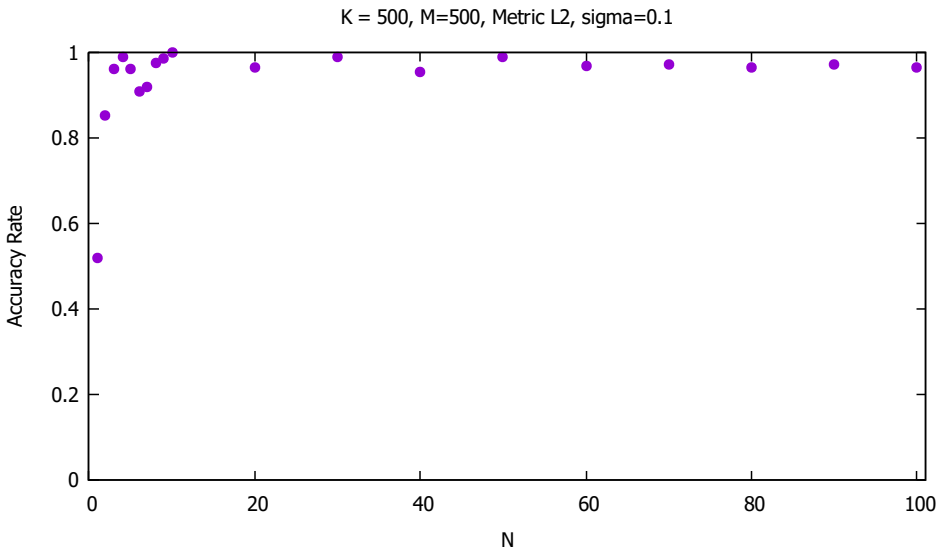

FIGURE 81

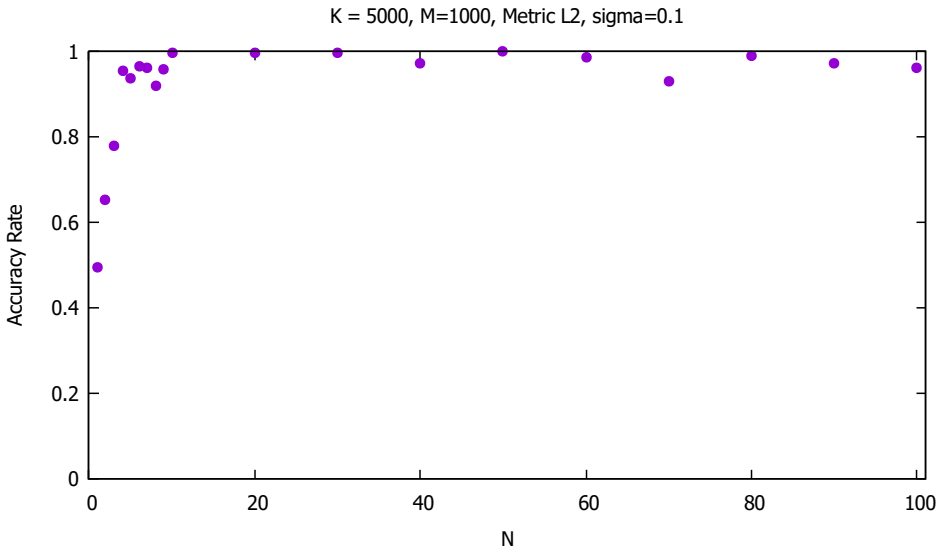

FIGURE 82

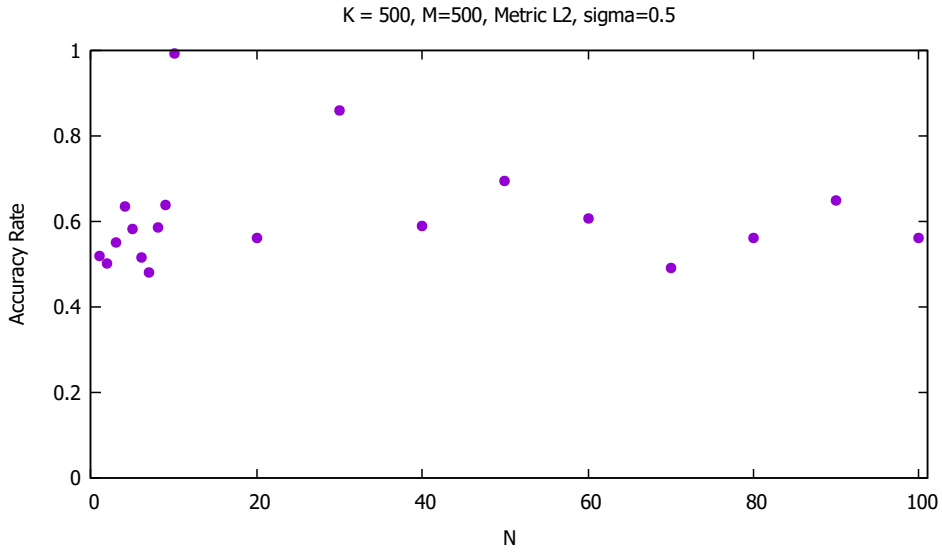

FIGURE 83

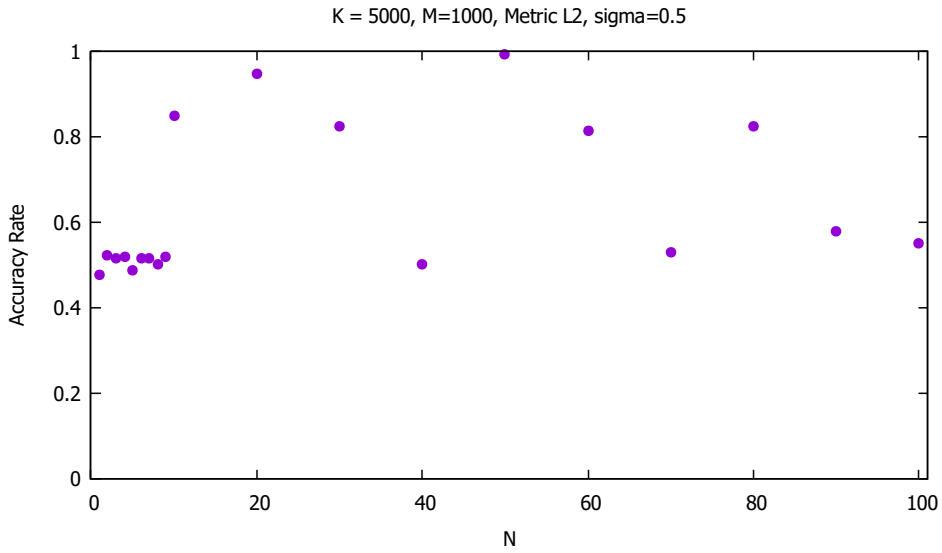

FIGURE 84

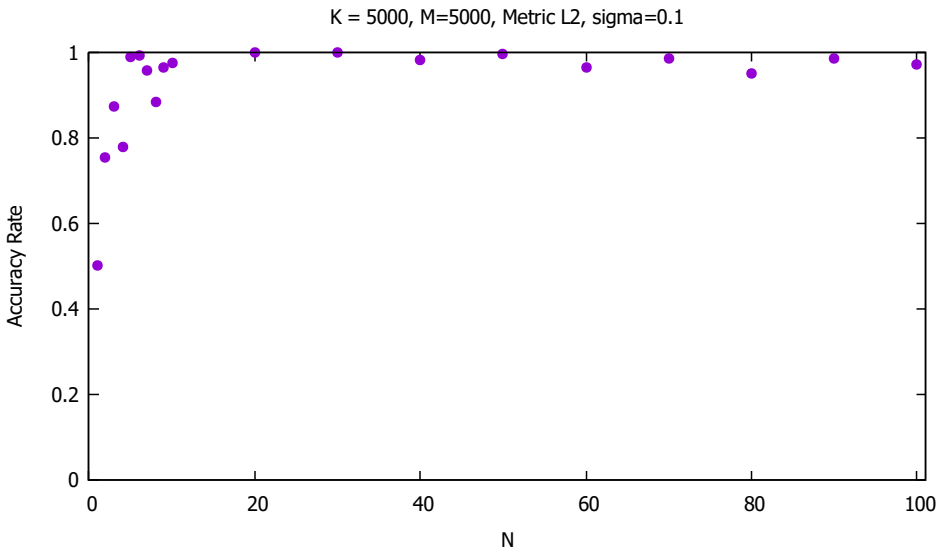
K = 5000, M=5000, Metric L2, sigma=0.1
Accuracy Rate
N
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100

FIGURE 85

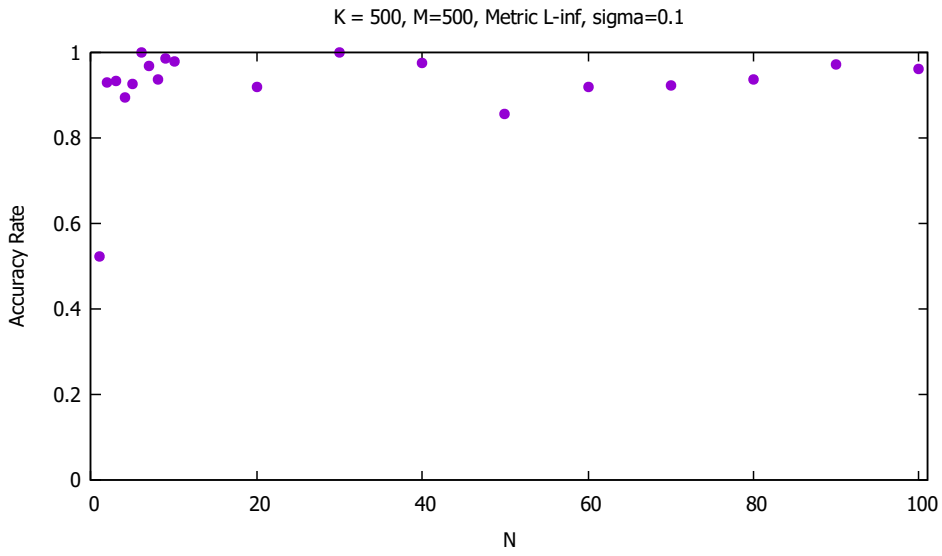

FIGURE 86

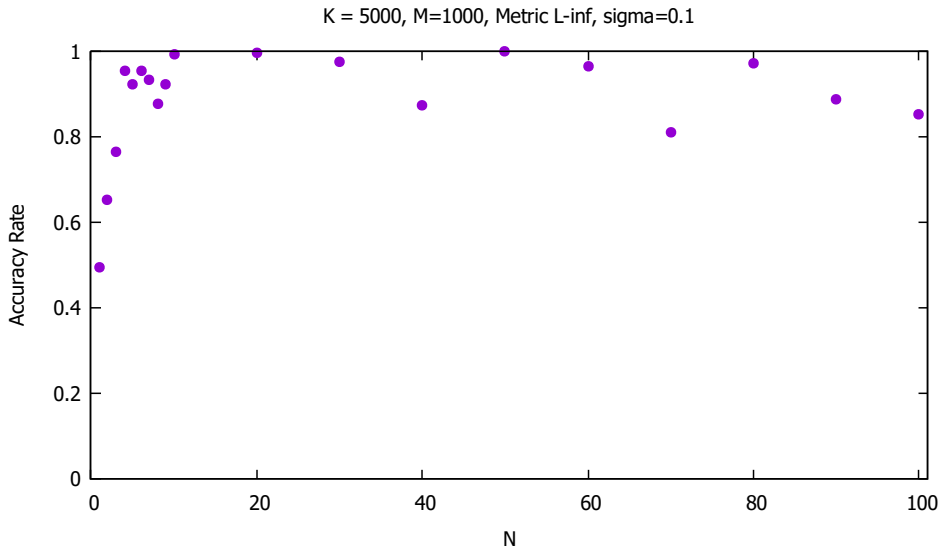

FIGURE 87

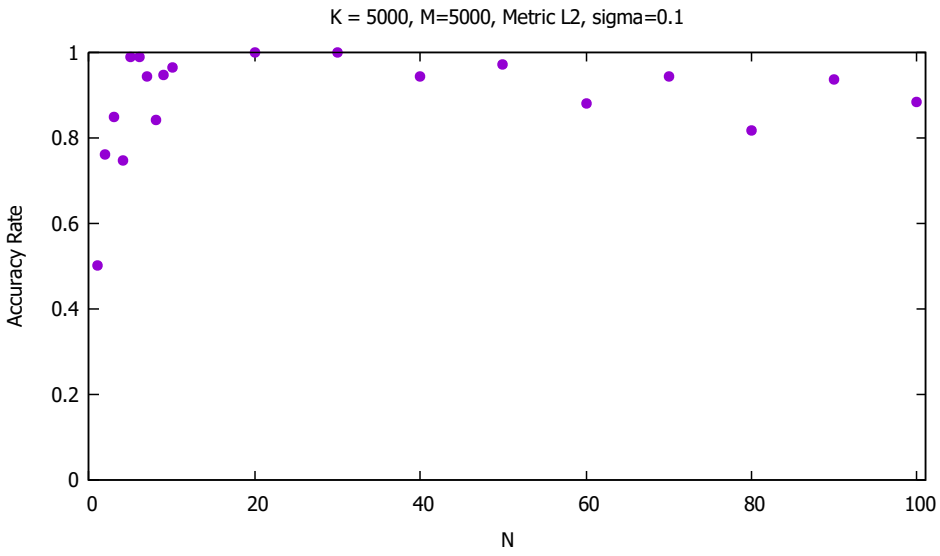
K = 5000, M=5000, Metric L2, sigma=0.1
Accuracy Rate
N
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100

FIGURE 88

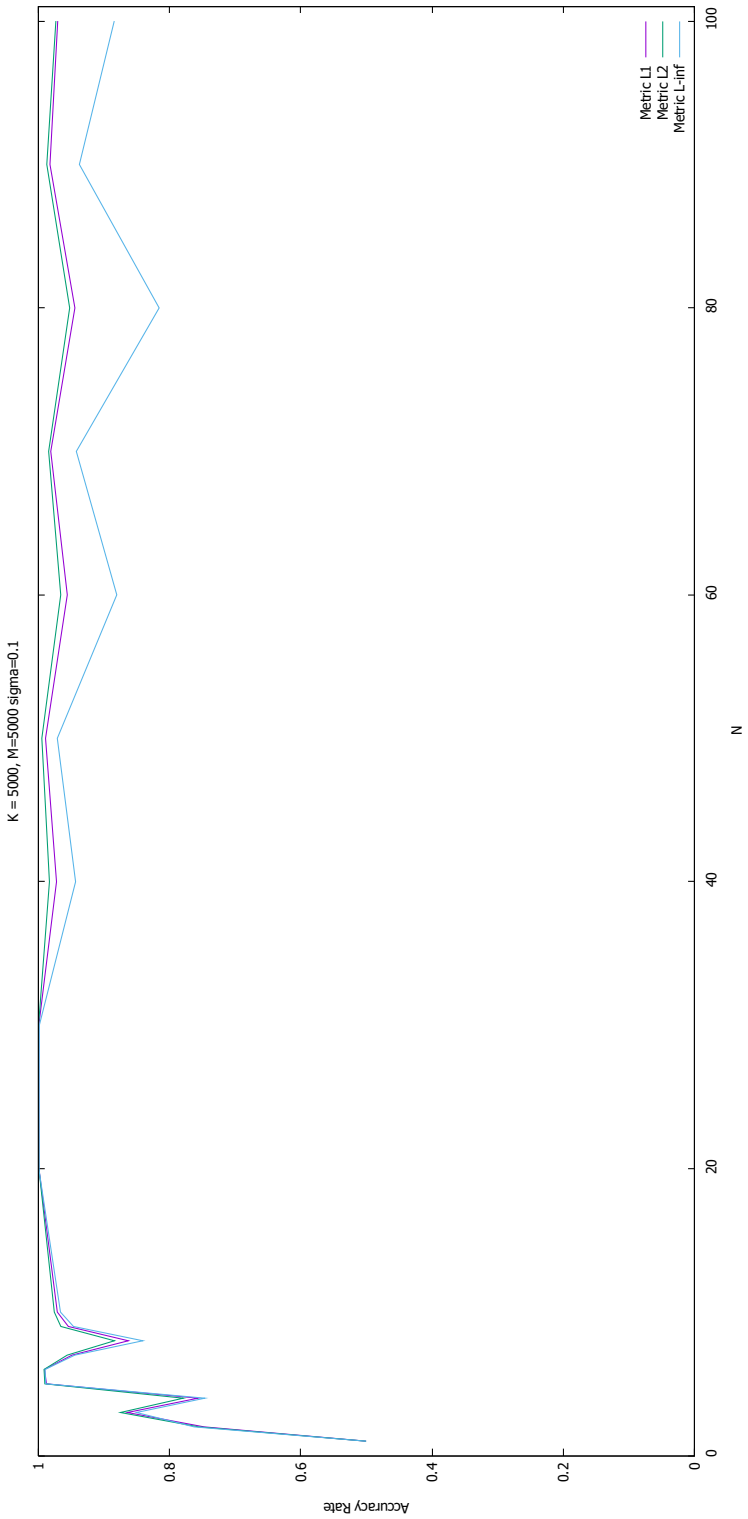
K = 5000, M=5000 sigma=0.1
Accuracy Rate
N
Metric L1
Metric L2
Metric L-inf
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100

FIGURE 89

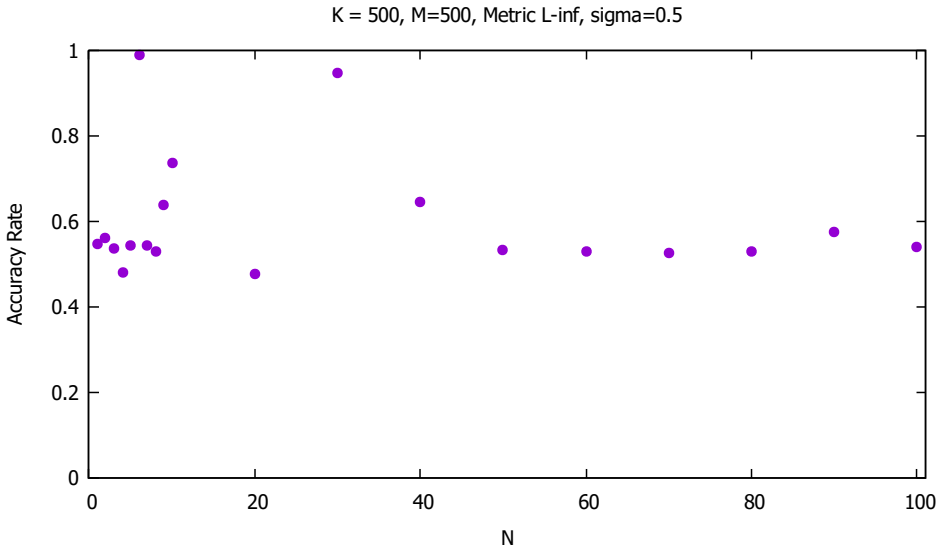

FIGURE 90

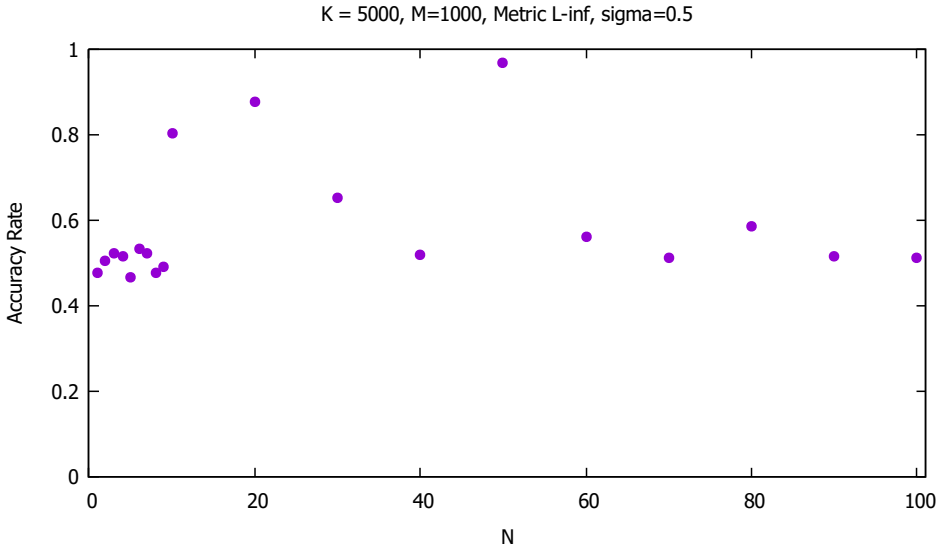

FIGURE 91

Appendix C. Graphs For Project 2 with Max Min Normalization

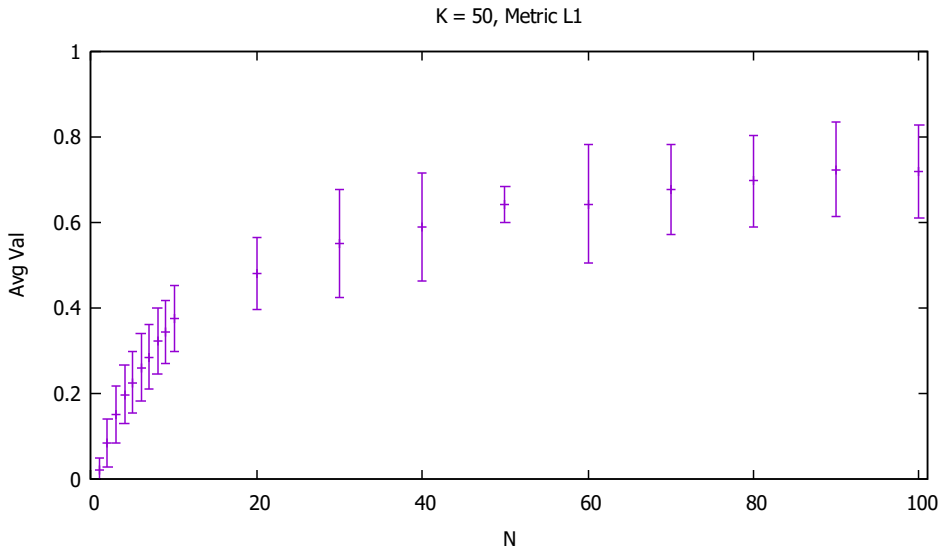

Figure 92

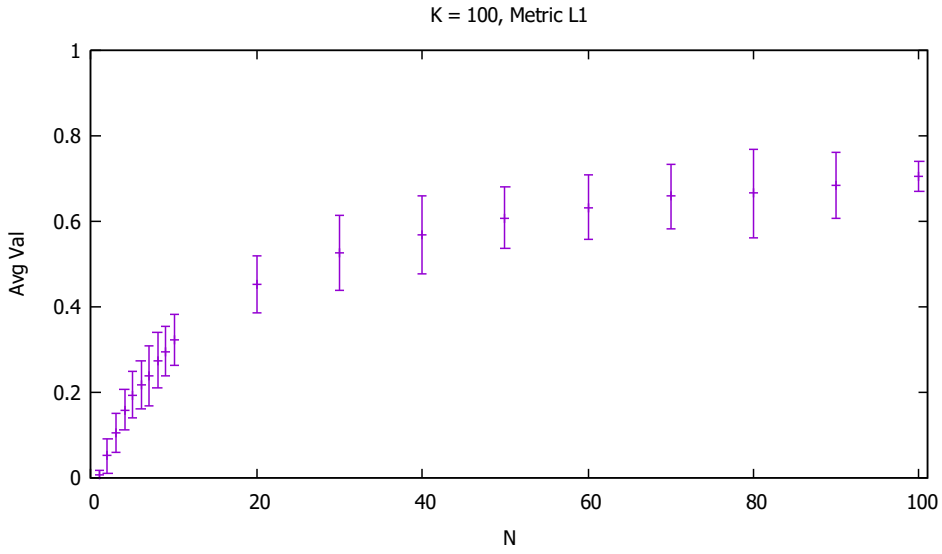

Figure 93

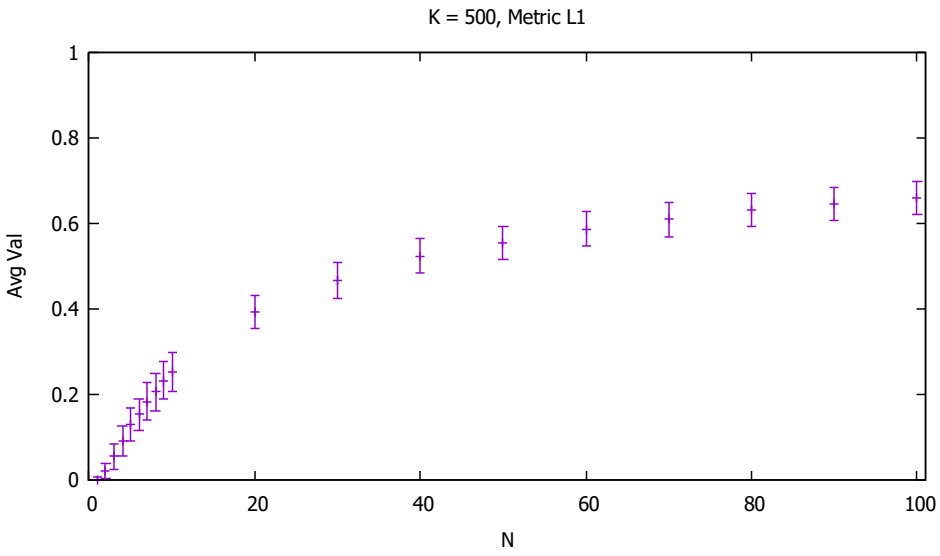

FIGURE 94

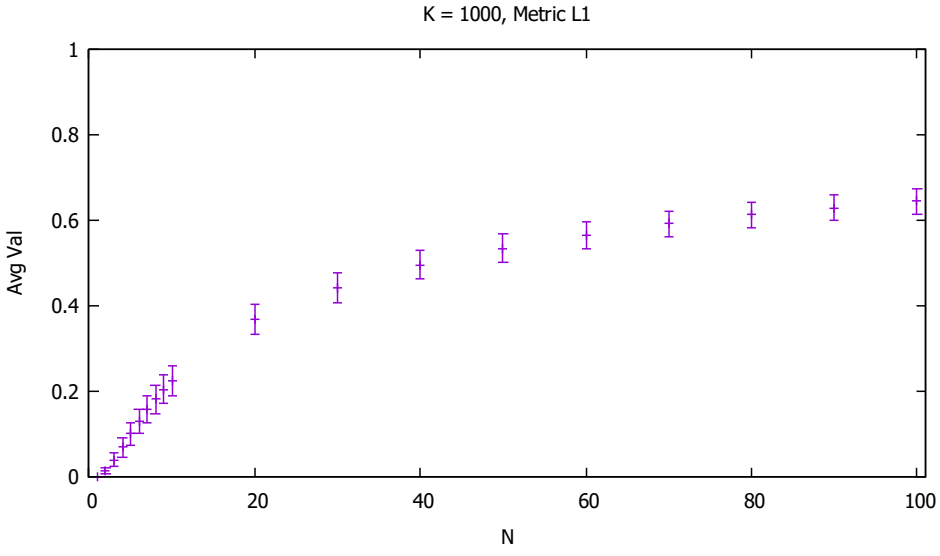

FIGURE 95

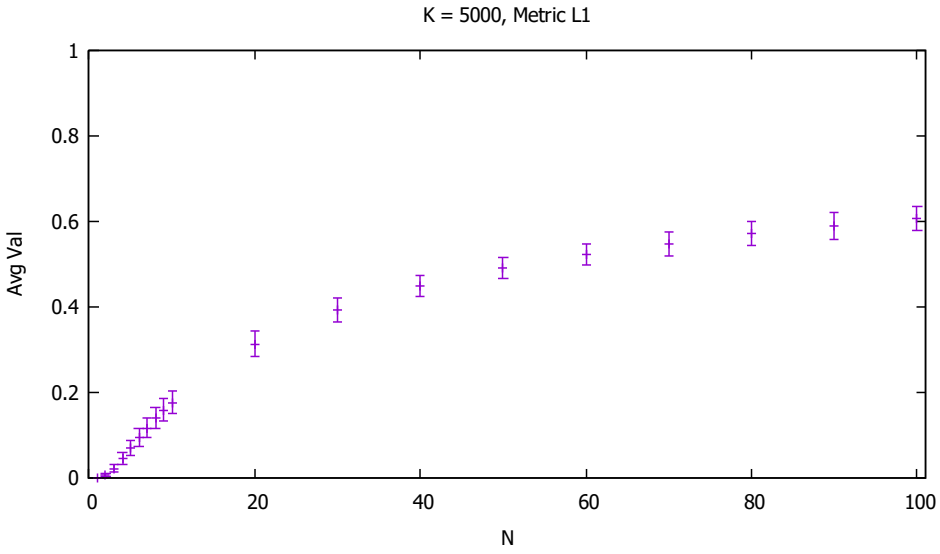

FIGURE 96

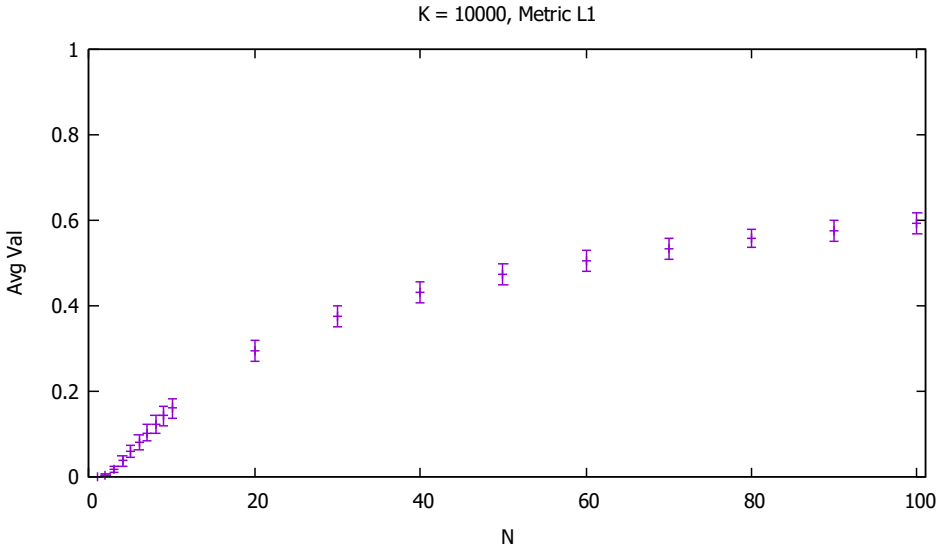

FIGURE 97

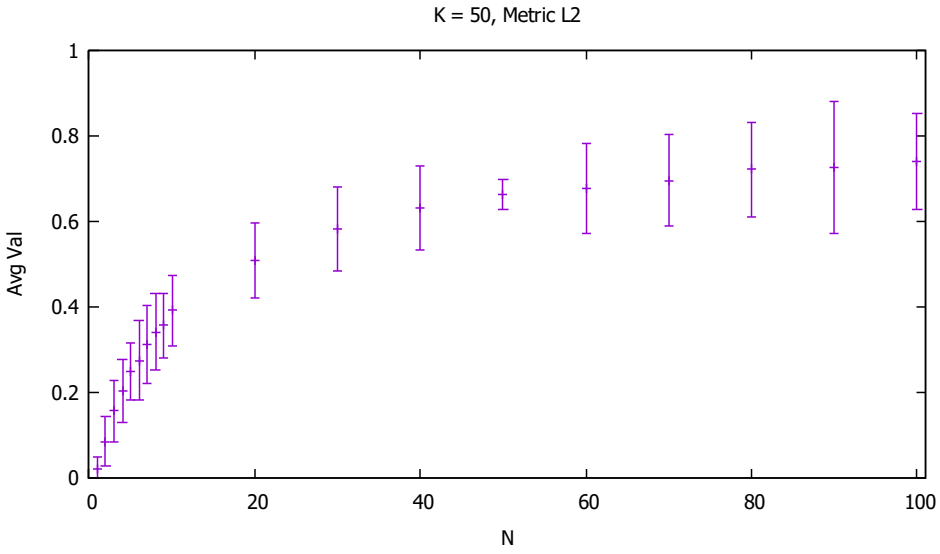

FIGURE 98

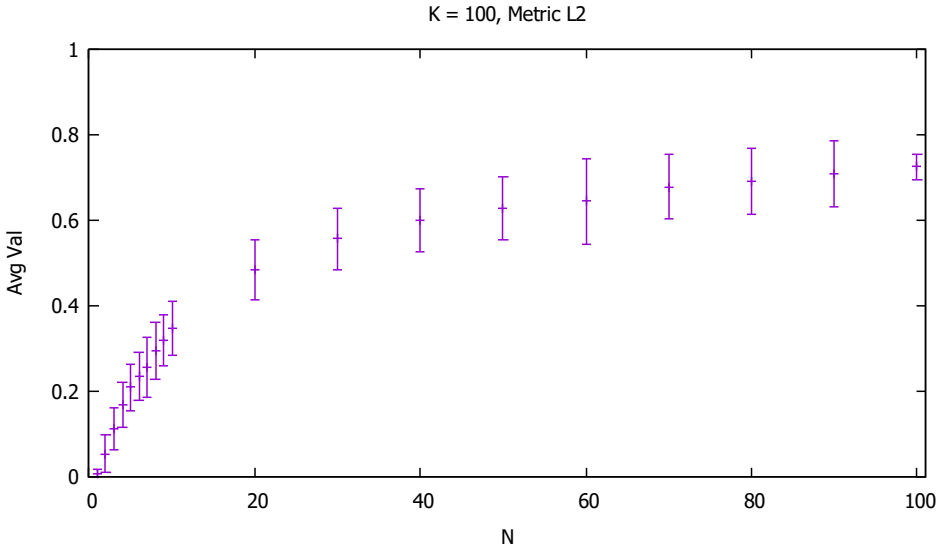

FIGURE 99

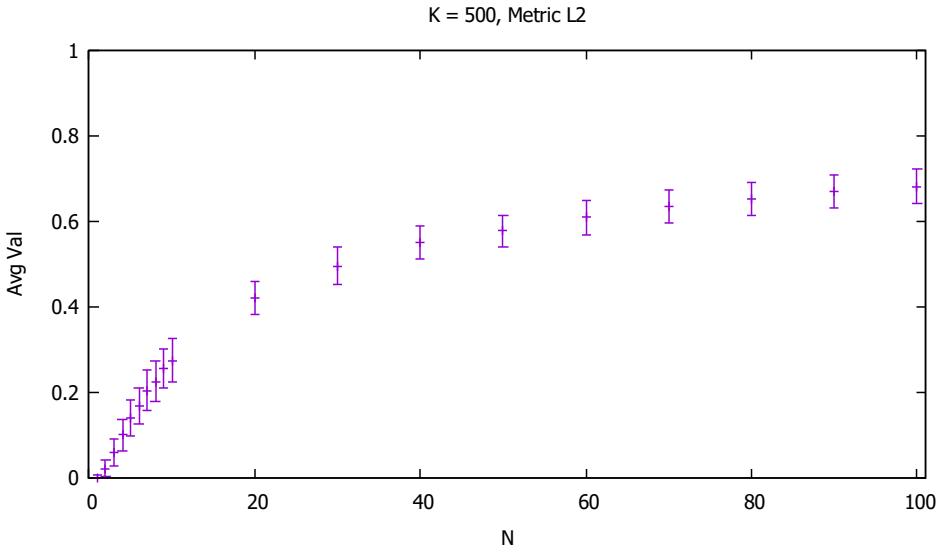

FIGURE 100

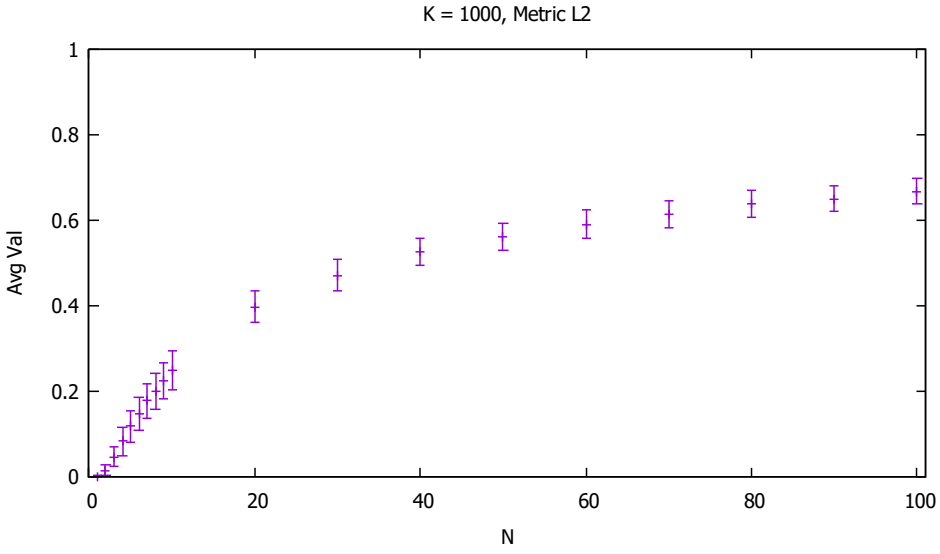

FIGURE 101

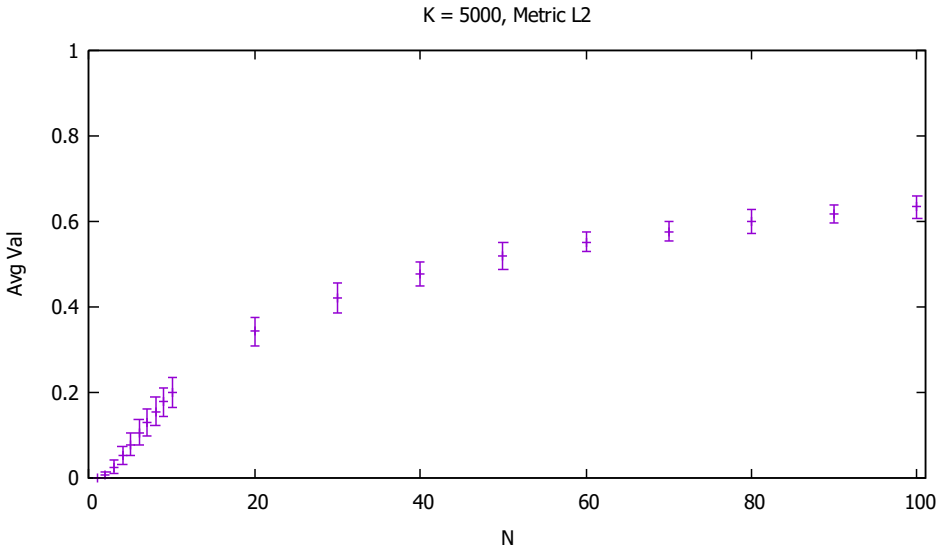

FIGURE 102

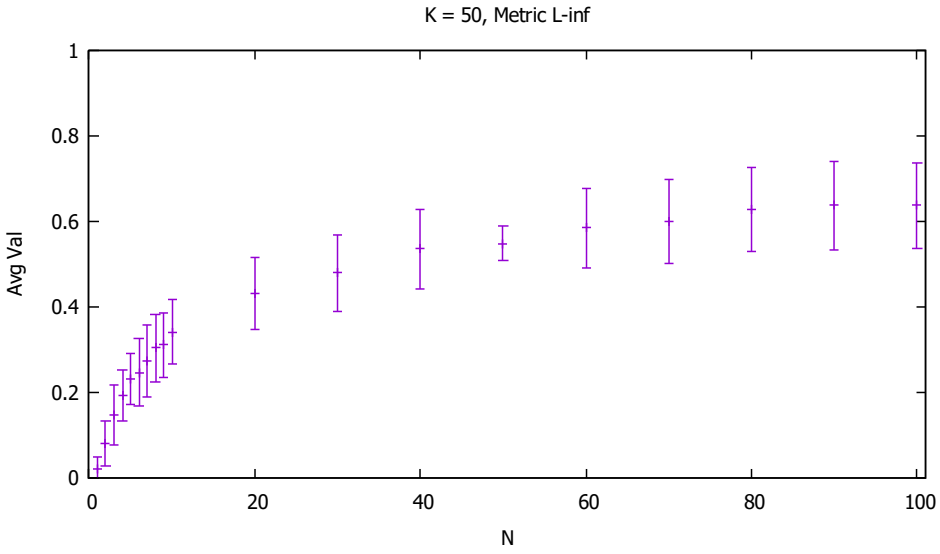

FIGURE 103

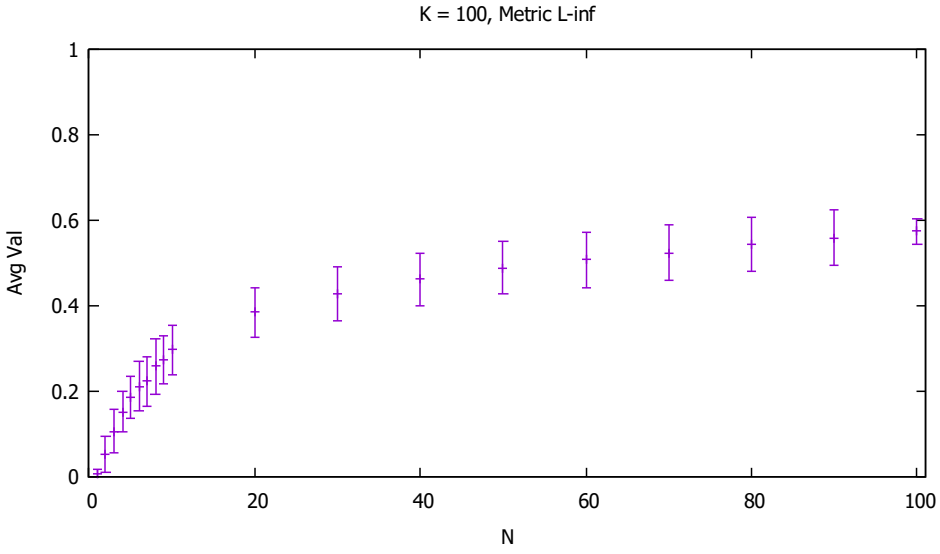

FIGURE 104

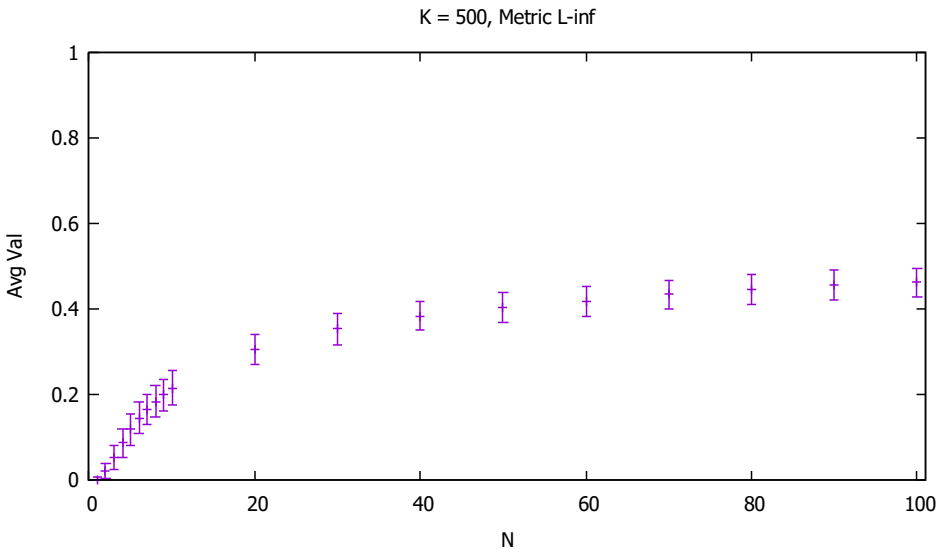

FIGURE 105

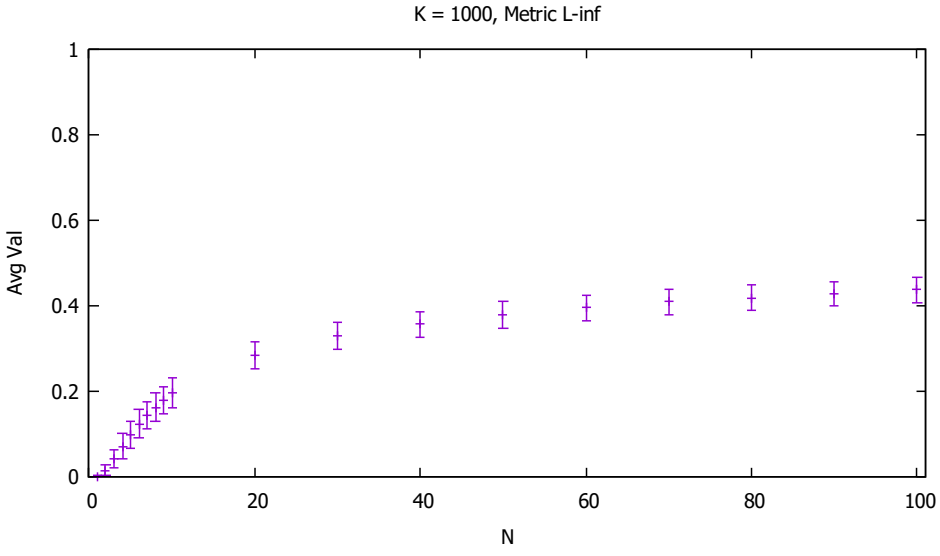

FIGURE 106

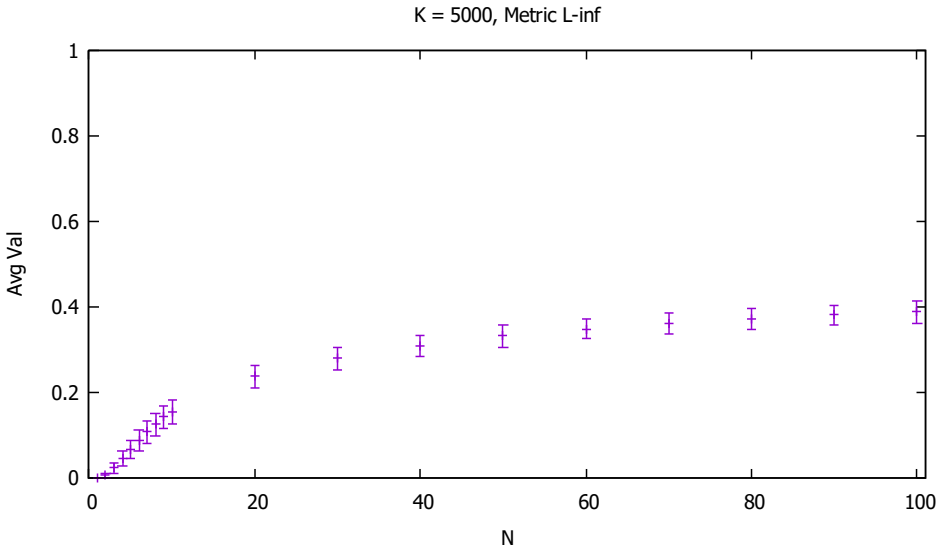

FIGURE 107

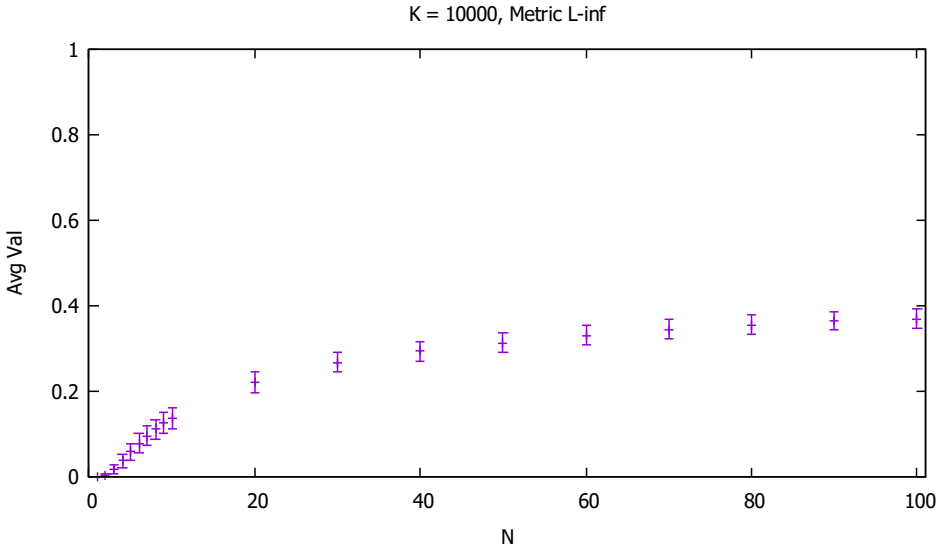

FIGURE 108

Appendix D. Graphs For Project 2 With Normalization With Respect To Euclidean Metric

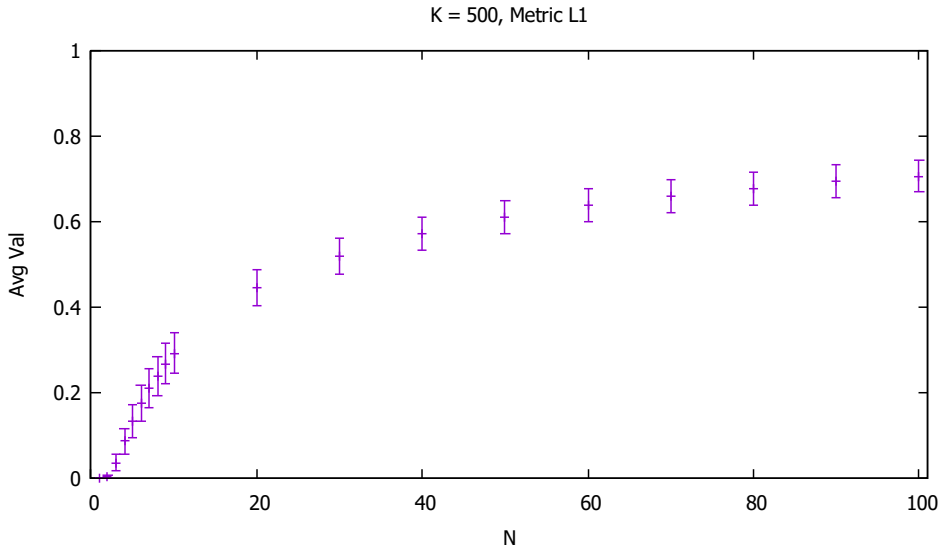

Figure 109

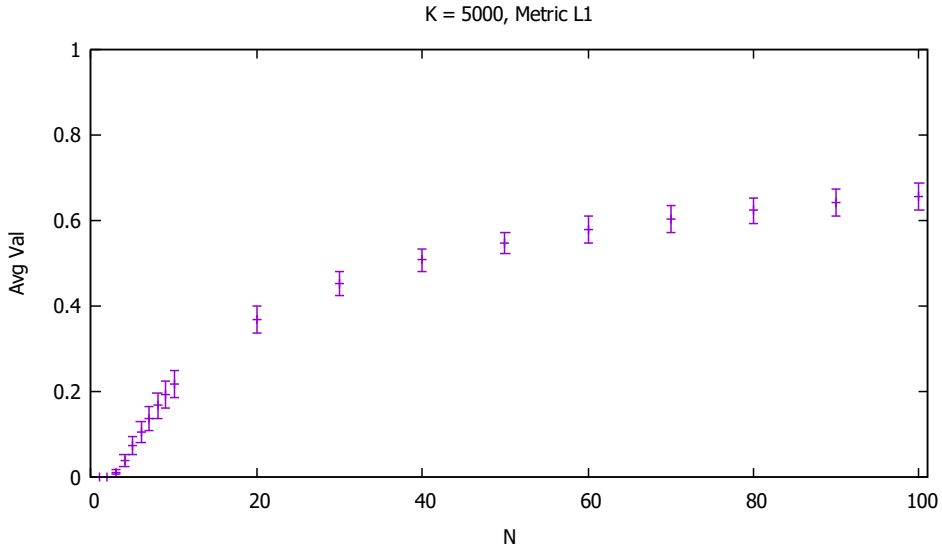

Figure 110

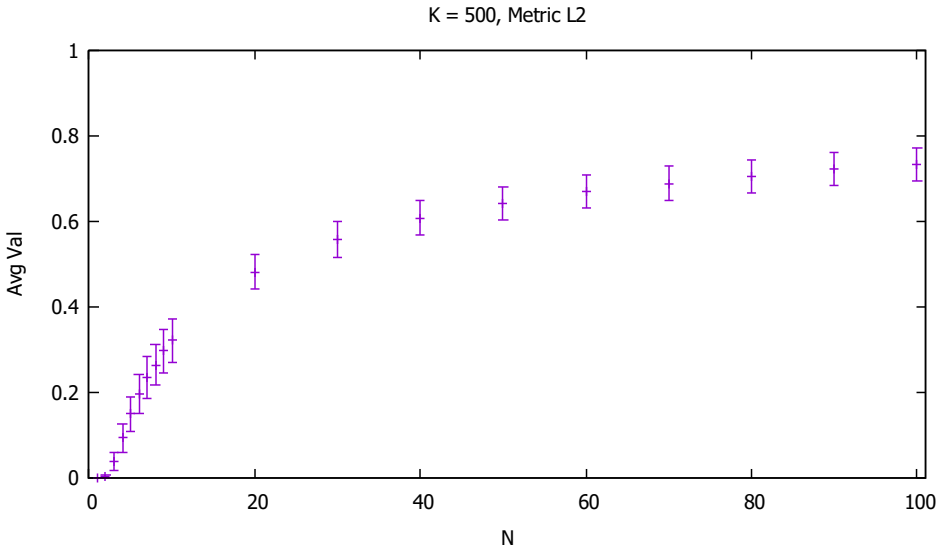

FIGURE 111

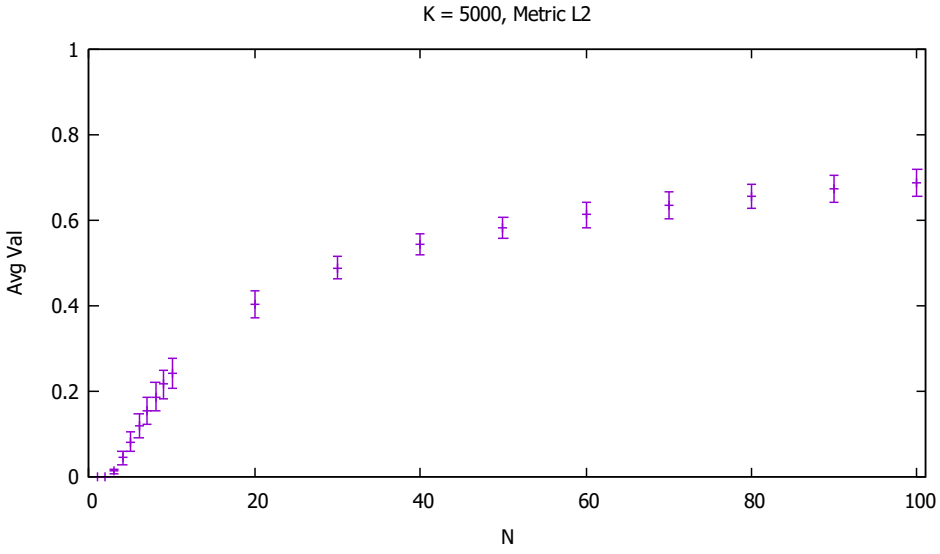

FIGURE 112

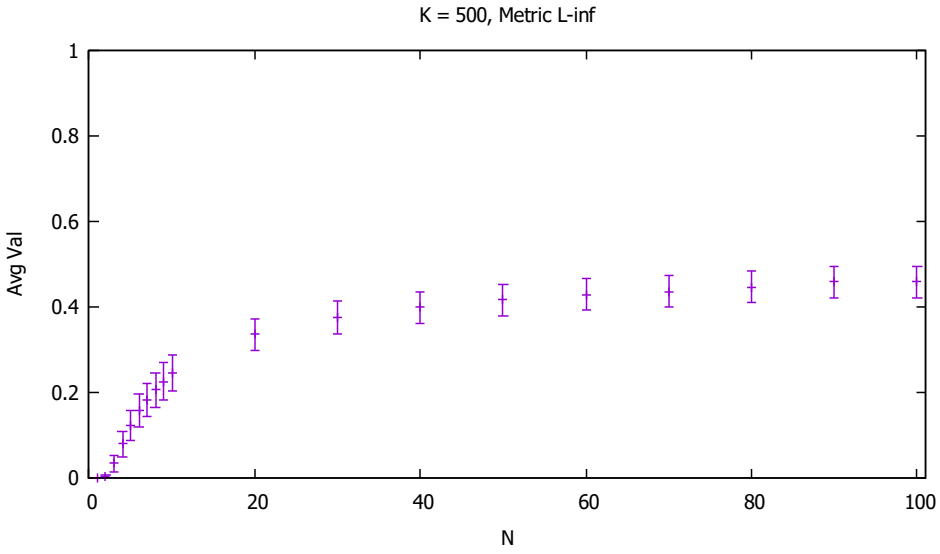

FIGURE 113

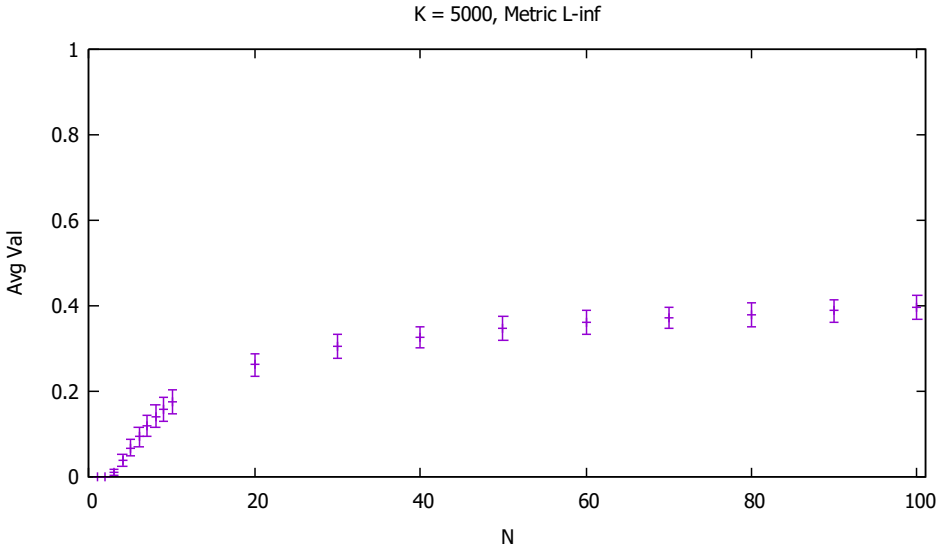

FIGURE 114

Appendix E. Comparison Graphs for Project 2

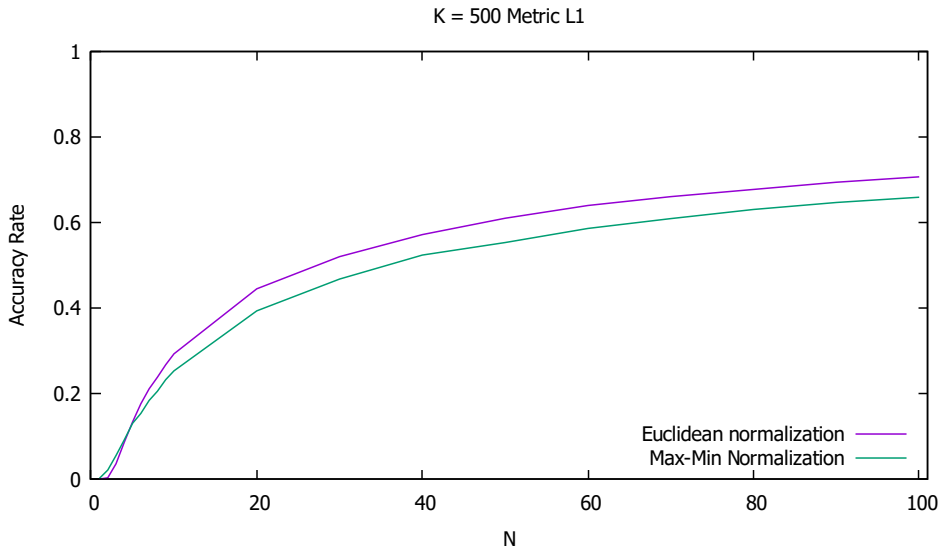

Figure 115

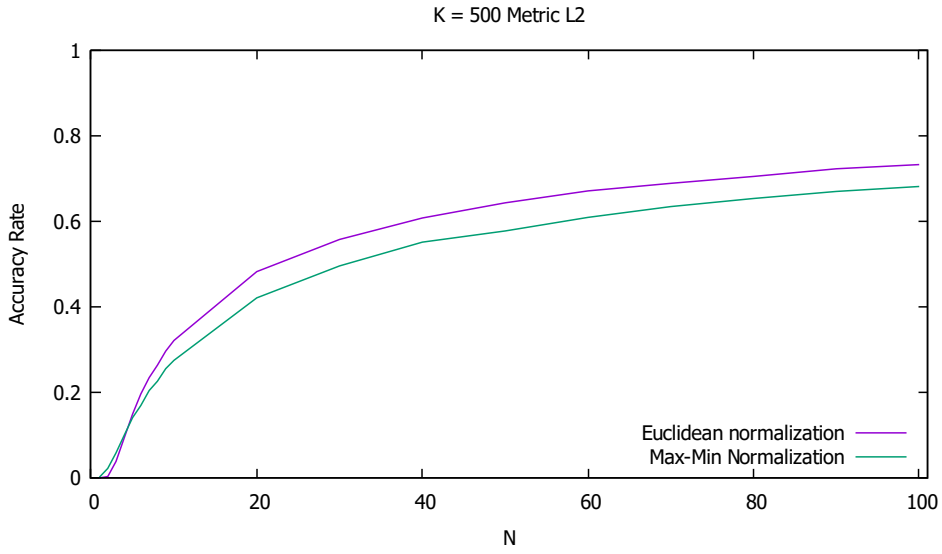

Figure 116

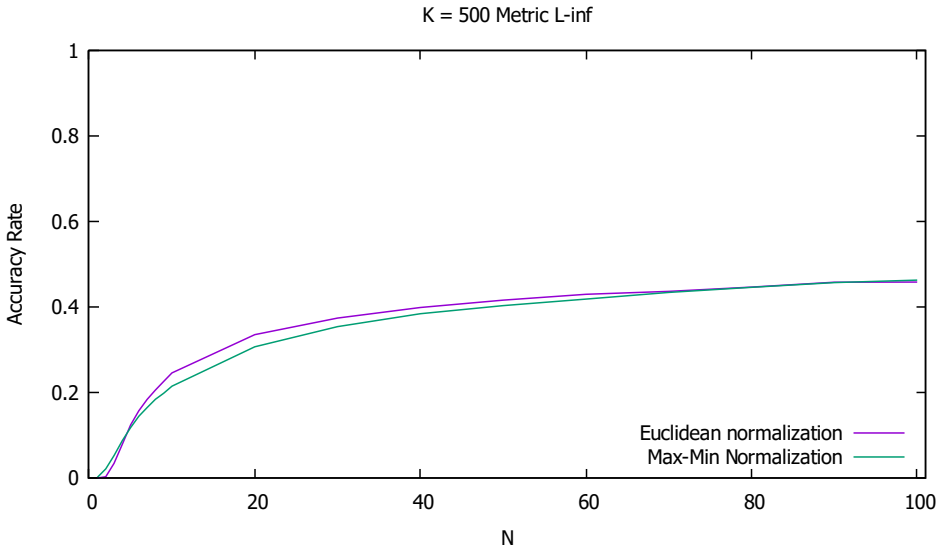

FIGURE 117

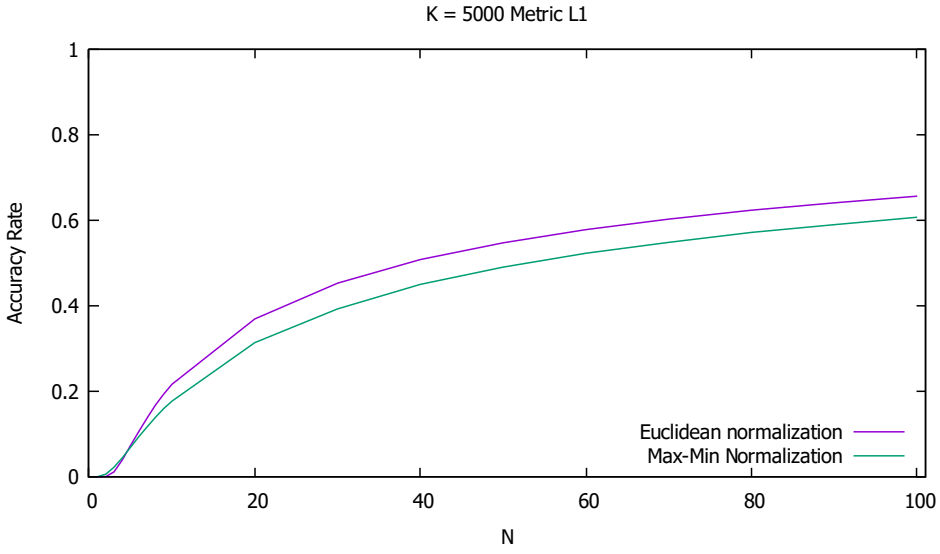

FIGURE 118

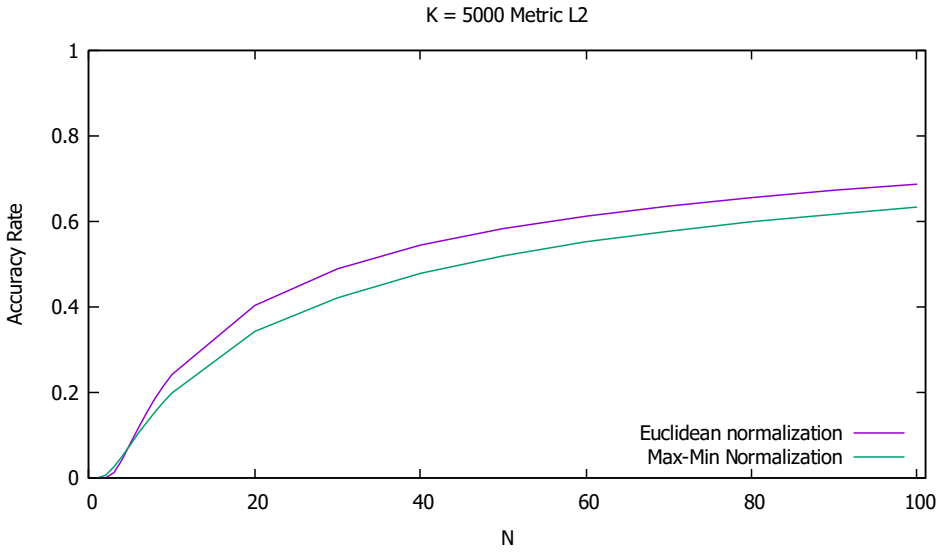

FIGURE 119

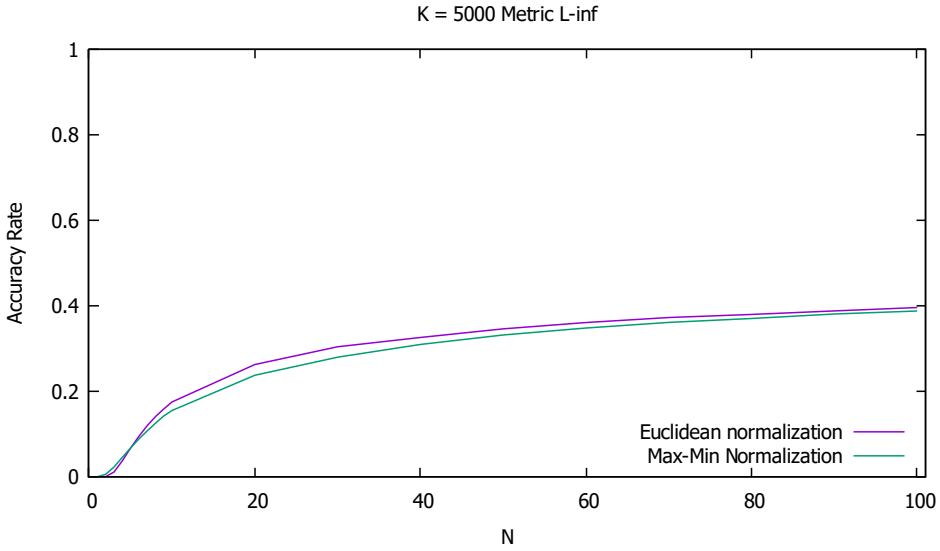

FIGURE 120