Minkowski's Metrics-Based k-NN Classifier Algorithm: A Comparative Study

Rolando Flores Carapia, Cornelio Yáñez Márquez

Pattern Recognition and Image Processing Laboratory
Center for Computing Research, National Polytechnic Institute, Mexico City, Mexico
rfcarapia@yahoo.com, cyanez@cic.ipn.mx

Abstract. The purpose of this paper is to present a comparative study of the use of three different Minkowski´s metrics in the k-NN (k Nearest Neighbor algorithm) classifier's design and operation: metric 1, euclidean and infinite. In a certain way, this paper is an experimental confirmation of the theoretical concepts suggested in [5], aiming at finding an optimal metric for the design and operation of the k-NN classifier. In this paper the k-NN algorithm is applied to one of the databases that has been most used in testing pattern recognition and machine learning algorithms, the *Iris* Plant Database. Our experiments confirm the hypothesis presented by the authors of [5], in the sense that euclidean metric tends to be the optimal metric for the k-NN algorithm.

1 Introduction

The k-NN algorithm (k Nearest Neighbor) is one of the longest-lived algorithms in the area of pattern classification, its simplicity making it attractive to researchers dedicated to pattern recognition and machine learning. As Cover and Hart [1] mention, the first formulation of a nearest neighbor rule was presented by Fix and Hodges in the early fifties. The nearest neighbor rule requires a set of labeled patterns, also known as fundamental set, where the class to which each pattern belongs is specified. This fundamental set is used to classify a set of test patterns, accounting for the distances from a test pattern of unknown class to each and every pattern in the fundamental set; assigning to the test pattern the class of the nearest pattern, according to a metric or function of distance previously chosen [9-11].

In the k-NN method the distance from a test pattern to each element of the fundamental set is calculated, the distances are sorted from lesser to greater and the majority rule is applied to the classes of those nearest k patterns, being k an arbitrary integer constant. This process is expensive in computational terms, for a relatively large fundamental set [2].

The importance of the metric or function of distance used in designing a k-NN classifier system is such that many researchers have approached the problem of finding the metric which yields the best results; that is, a statistically optimal metric [3-8].

© A. Gelbukh, C. Yáñez Márquez, O. Camacho Nieto (Eds.)
Advances in Artificial Intelligence and Computer Science
Research on Computing Science 14, 2005, pp. 191-202

Short and Fukunaga [3] show a local distance measure to optimize the performance of the 2-NN classifier for a finite number of samples, and point that the close relationship between NN classification and density estimation suggests that an analogous performance improvement in classification error may also be obtained by a proper choice of a distance measure.

In a specific application area, character recognition, Waard [4] proposes a simple selective learning method using an optimised distance, and he discusses a fast search method. In another related work, Imiya [6] defines a distance measure among spatial lines, and by using the geodesic distance and the proposed metric, he constructs a method for the classification of spatial lines. In more recent works, Jin and Kurniawati [7] try using different metrics in searches related to visual information retrieval, and Peng *et. al.* [8] propose an adaptive nearest neighbor classification method which allows minimizing the bias caused by the variability of class conditional probabilities when high dimention patterns are used.

The current paper is inspired by the theoretical concepts suggested by Snapp y Venkatesh [5], who analyze the Minkowski's metrics in the k-NN classifier's design and operation, pointing out that the optimal metric approaches the euclidean as the size of the fundamental set grows. This work is, in a certain way, an experimental confirmation of the theoretical results of Snapp and Venkatesh, by applying the k-NN algorithms to a very popular database.

The rest of the paper is organized as follows: in section 2 Minkowski's norms and metrics are introduced, since they are the basis for the algorithms designed during the experimental study. Section 3 is dedicated to presenting the k-NN algorithm, both for when k = 1 as for when k is a positive integer number greater than 1; it also includes an illustrative example. Section 4 contains the proposed methodology which allows to perform the comparative study, while section 5 is the essential part of the paper, showing the experimental results. The conclusions are presented in section 6, along with suggestions for future work, finalizing with acknowledgements and references.

2 Minkowski's Norms and Metrics

This section is dedicated to presenting Minkowski's norms and metrics, since they are used in the algorithms designed during the experimental study. The concepts and definitions presented in this section were taken from [10] and [11]. First the concept of norm is defined, so as to then derive the definition of Minkowski's norms and later do something similar with the concept of metric and the definition of Minkowski's metrics

2.1 Norms

A norm on R^n is a function $N : R^n \to R$ which complies with the following propierties:

$$N(x) \geq 0, \quad \forall x \in R^n \tag{1}$$

$$N(\alpha x) = |\alpha| N(x), \qquad x \in R^n, \alpha \in R \tag{2}$$

$$N(x) = 0 \text{ if and only if } x = 0, \qquad x \in R^n \tag{3}$$

$$N(x+y) \leq N(x) + N(y) \qquad \forall x, y \in R^n \text{ (Triangle inequality)} \tag{4}$$

2.1.1 Minkowski's Norms

Let $x \in R^n$, the Minkowski's norm of order p with $p \in Z^+$ is defined as:

$$\|x\|_p = \left[\sum_{i=1}^{n} |x_i|^p\right]^{\frac{1}{p}} \text{ donde } x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{5}$$

There are three particular cases of Minkowski's norm:

Norm 1: $p = 1$

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|^1 = |x_1| + |x_2| + |x_3| + \cdots + |x_n| \tag{6}$$

Euclidean Norm: $p = 2$

$$\|x\|_2 = \left[\sum_{i=1}^{n} |x_i|^2\right]^{\frac{1}{2}} = \sqrt[2]{|x_1|^2 + |x_2|^2 + |x_3|^2 + \cdots |x_n|^2} \tag{7}$$

Infinite Norm: $p \to \infty$

$$\|x\|_\infty = \bigvee_{i=1}^{n} |x_i| = m\acute{a}x\{x_1|, |x_2|, |x_3|, \cdots, |x_n|\} \tag{8}$$

2.2 Metric

Even though we have the intuitive notion that a metric is a way of measuring a distance, its formal definition is as follows:

A metric on R^n is a function $\rho: R^n \times R^n \to R$ which satisfies the following propierties:

$$\rho(x,y) = 0 \text{ if and only if } x = y, \quad \forall x,y \in R^n \tag{9}$$

$$\rho(x,y) = \rho(y,x), \quad \forall x,y \in R^n \tag{10}$$

$$\rho(x,y) \le \rho(x,z) + \rho(z,y), \quad \forall x,y,z \in R^n \tag{11}$$

Generally speaking, there is an infinite number of functions that fulfill the former definition, and therefore the distance between two points calculated by a metric can be completely different to the distance calculated by another metric.

2.2.1 Minkowski's Metrics

Minkowski's metrics are induced by Minkowski's norms:

$$\|x-y\|_p = \left[\sum_{i=1}^{n} |x_i - y_i|^p\right]^{\frac{1}{p}} \quad x,y \in R^n, \quad p,n \in Z^+ \tag{12}$$

The metric will behave according to p. There are three values that are most commonly used:

Metric 1: $p = 1$

$$\|x-y\|_1 = \sum_{i=1}^{n} |x_i - y_i| \quad x,y \in R^n \tag{13}$$

Euclidean Metric: $p = 2$

$$\|x-y\|_2 = \left[\sum_{i=1}^{n} |x_i - y_i|^2\right]^{\frac{1}{2}} \quad x,y \in R^n \tag{14}$$

Infinite Metric: $p \to \infty$

$$\|x-y\|_\infty = \bigvee_{i=1}^{n} |x_i - y_i| \quad x,y \in R^n \tag{15}$$

3 The k-NN Classifier

In this section the k-NN algorithm is presented, both for when k = 1 as for when k is a positive integer number greater than 1; it also includes an illustrative example. The contents of this section is fundamentally based on [1], [2] and [11].

3.1 The 1-NN Classifier Algorithm

Given the following finite fundamental set of ρ pairs of patterns, with ρ being a positive integer, being x^{μ} a n-dimensional real pattern and y^{μ} the assigned class to this pattern:

$$\{(x^{\mu}, y^{\mu}), \mu = 1,2,3,\cdots,\rho\}$$

the algorithms to design and operate a k-NN classifier are described next:

3.2 Algorithm For k = 1:

1 Select the metric.
2 Let x be a pattern to classify. Calculate the distances to each and every pattern in the fundamental set.
3 Obtain the minimal distance.
4 Assign to x the class of the pattern with the minimal distance.

3.3 Algorithm For k > 1:

1 Select the metric.
2 Let x be a pattern to classify. Calculate the distances to each and every pattern in the fundamental set.
3 Sort the data in ascending order.
4 Obtain the k lesser values of distance.
5 Use the *majority rule* to assign the class.

A significant fact to mention is the metric to be used in the algorithm. Obviously the result of the distances will depend on the metric used, being the Minkowski's metrics the most widely used [5].

3.4 Example

Let $\{(x^1, y^1), (x^2, y^2), (x^3, y^3)\}$ be the fundamental set.

Where: $x^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, y^1 = \omega_1, \quad x^2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, y^2 = \omega_2, \quad x^3 = \begin{pmatrix} 0 \\ 5 \end{pmatrix}, y^3 = \omega_3$

And let: $x^4 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ be a pattern to classify.

1.- For this example, we select the euclidean metric.
2.- Using equation 14 (euclidean metric), we calculate the distances to each element of the fundamental set.

$$d_1 = \left[\sum_{i=1}^{2} \left| x_i^1 - x_i^4 \right|^2 \right]^{\frac{1}{2}} = \sqrt[2]{(1-2)^2 + (1-2)^2} = \sqrt[2]{2}$$

$$d_2 = \left[\sum_{i=1}^{2} \left| x_i^2 - x_i^4 \right|^2 \right]^{\frac{1}{2}} = \sqrt[2]{(2-2)^2 + (1-2)^2} = \sqrt[2]{1}$$

$$d_3 = \left[\sum_{i=1}^{2} \left| x_i^3 - x_i^4 \right|^2 \right]^{\frac{1}{2}} = \sqrt[2]{(0-2)^2 + (5-2)^2} = \sqrt[2]{13}$$

3.- Since $d_2 < d_1$ y $d_2 < d_3$, d_2 is the minimal distance.

4.- Since d_2 was obtained from the element x^2 and it belongs to the class ϖ_2, we may conclude that x^4 belongs to the class ϖ_2.

4 Proposed Methodology

Given that our purpose is to test the k-NN classifier's algorithm with Minkowski's metrics for different values of p, over trustworthy and proven data, we have decided to chose the *Iris* Plant Database, available at the UCI Machine Learning Repository site [13]. This is one of the most popular and trustworthy databases, and has been used to test an important number of pattern classifying algorithms. It was ellaborated by E. Anderson and initially presented by Fisher [12] in 1936. The Iris Plant Database has 150 data patterns extracted from three classes of flowers: Iris setosa, Iris virginica and Iris versicolor, having each class 50 representative patterns, and each pattern having four features: sepal lenght, sepal, width, petal leght, and petal width. In Table I is shown an example of each class, as contained in the database. This database con be consulted in its complete state in [12] or be downloaded from [13].

Table I. Examples from the Iris Plant Database

sepal length	sepal width	petal length	petal width	class
5.1	3.5	1.4	0.2	Iris setosa
...	...	...	...	...
7.0	3.2	4.7	1.4	Iris versicolor
...	...	...	...	...
6.3	3.3	6.0	2.5	Iris virginica

In order to test the k-NN algorithm we generated a computer program that reads the UCI's Iris Plant Database [12], forming a fundamental set with 90 of its patterns and classifying the rest with respect to such fundamental set.

The program's algorithm is the following:

```
Read the Iris Plant Database
For i = 1 until 100
    Do
        Take 30 random elements from each class (Iris setosa, Iris versicolor
        and Iris virginica).
        Execute the k-NN algorithm with k = 1 and the 1 metric.
        Execute the k-NN algorithm with k = 1 and the euclidean metric.
        Execute the k-NN algorithm with k = 1 and the infinite metric.
    End Do
End For
Show how many patterns were correctly and incorreclty classified in a table
```

The analysis of the results will show which of the Minkowski's metrics yields the best results and, therefore, approaches Snapp and Venkatesh's hypothesis presented in [5].

5 Experimental Results

In Table II are shown the results obtained when executing the program. As can be seen, the results given by metric 1 and the euclidean metric are quite alike, except in some cases in which the euclidean metric performs better classifications.

Table 1. Results Table

Iteration	Metric 1		Euclidean Metric		Infinite Metric	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
1	57	03	57	03	33	27
2	57	03	57	03	41	19
3	57	03	58	02	38	22
4	59	01	59	01	36	24
5	57	03	57	03	38	22
6	57	03	57	03	45	15
7	57	03	57	03	43	17
8	58	02	58	02	31	29
9	57	03	57	03	47	13
10	56	04	56	04	35	25
11	57	03	57	03	37	23
12	58	02	58	02	39	21
13	57	03	57	03	38	22
14	57	03	57	03	40	20
15	57	03	57	03	45	15
16	58	02	59	01	41	19

Iteration	Metric 1		Euclidean Metric		Infinite Metric	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
17	57	03	58	02	47	13
18	58	02	58	02	40	20
19	56	04	56	04	42	18
20	58	02	58	02	38	22
21	59	01	59	01	41	19
22	59	01	59	01	43	17
23	57	03	56	04	41	19
24	59	01	59	01	38	22
25	59	01	59	01	38	22
26	58	02	59	01	40	20
27	57	03	57	03	46	14
28	55	05	56	04	44	16
29	56	04	56	04	39	21
30	56	04	56	04	43	17
31	56	04	58	02	41	19
32	57	03	58	02	38	22
33	58	02	58	02	42	18
34	55	05	56	04	46	14
35	58	02	57	03	41	19
36	57	03	57	03	39	21
37	58	02	58	02	36	24
38	57	03	58	02	38	22
39	56	04	57	03	44	16
40	57	03	57	03	44	16
41	55	05	55	05	40	20
42	56	04	58	02	37	23
43	58	02	58	02	45	15
44	58	02	58	02	42	18
45	58	02	59	01	44	16
46	59	01	59	01	40	20
47	55	05	55	05	39	21
48	55	05	58	02	40	20
49	58	02	58	02	44	16
50	58	02	58	02	43	17
51	55	05	56	04	40	20
52	57	03	57	03	41	19
53	56	04	57	03	40	20
54	57	03	58	02	38	22
55	58	02	58	02	44	16
56	53	07	54	06	40	20
57	57	03	57	03	37	23
58	58	02	58	02	36	24
59	57	03	57	03	41	19
60	55	05	56	04	44	16
61	57	03	57	03	32	28
62	57	03	58	02	32	28
63	58	02	59	01	49	11
64	56	04	56	04	43	17
65	58	02	58	02	40	20
66	56	04	58	02	45	15
67	57	03	58	02	41	19
68	54	06	55	05	47	13
69	59	01	59	01	43	17

Iteration	Metric 1		Euclidean Metric		Infinite Metric	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
70	56	04	56	04	35	25
71	58	02	58	02	40	20
72	57	03	57	03	42	18
73	57	03	57	03	38	22
74	56	04	56	04	44	16
75	57	03	57	03	38	22
76	58	02	58	02	46	14
77	55	05	55	05	46	14
78	59	01	59	01	36	24
79	57	03	56	04	38	22
80	58	02	58	02	45	15
81	55	05	55	05	36	24
82	59	01	59	01	48	12
83	56	04	56	04	40	20
84	57	03	57	03	38	22
85	57	03	58	02	43	17
86	56	04	56	04	45	15
87	56	04	57	03	40	20
88	56	04	57	03	31	29
89	58	02	58	02	42	18
90	59	01	59	01	35	25
91	57	03	58	02	46	14
92	58	02	58	02	46	14
93	57	03	57	03	47	13
94	59	01	59	01	35	25
95	56	04	56	04	37	23
96	57	03	57	03	33	27
97	57	03	57	03	48	12
98	57	03	57	03	39	21
99	58	02	58	02	44	16
100	57	03	59	01	44	16

If we look closely at iteration 3 (in Table II), we can see there is a difference between metric 1 and the euclidean metric. This difference is due to the fundamental set containing the following elements:

$$\{(x^1, y^1), \cdots (x^i, y^i), \cdots (x^j, y^j), \cdots (x^{90}, y^{90})\}$$

Where: $x^i = \begin{pmatrix} 4.9 \\ 1.5 \\ 6.3 \\ 2.5 \end{pmatrix}, y^i = \varpi_2$ y $x^j = \begin{pmatrix} 5.3 \\ 1.9 \\ 6.4 \\ 2.7 \end{pmatrix}, y^j = \varpi_3$, when classifying the pattern $x^a = \begin{pmatrix} 5.6 \\ 1.4 \\ 6.1 \\ 2.6 \end{pmatrix}$ that belongs to the class ϖ_3 we have two cases of relevance:

a) Using metric 1 (equation [13]):

$$d_1 = \left\| x^a - x^i \right\|_1 = \sum_{k=1}^{n} \left| x_k^a - x_k^i \right| = \left| x_1^a - x_1^i \right| + \left| x_2^a - x_2^i \right| + \left| x_3^a - x_3^i \right| + \left| x_4^a - x_4^i \right|$$

$$d_2 = \left\| x^a - x^j \right\|_1 = \sum_{i=1}^{n} \left| x_k^a - x_k^j \right| = \left| x_1^a - x_1^j \right| + \left| x_2^a - x_2^j \right| + \left| x_3^a - x_3^j \right| + \left| x_4^a - x_4^j \right|$$

$$d_1 = |5.6 - 4.9| + |1.4 - 1.5| + |6.1 - 6.3| + |2.6 - 2.5| = 0.7 + 0.1 + 0.2 + 0.1 = 1.1$$

$$d_2 = |5.6 - 5.3| + |1.4 - 1.9| + |6.1 - 6.4| + |2.6 - 2.7| = 0.3 + 0.5 + 0.3 + 0.1 = 1.2$$

Since $d_1 < d_2$, the pattern x^a belongs to the class ϖ_2 which is an error, given that it belongs to the class ϖ_3.

b) Using the euclidean metric (equation [14]):

$$d_1 = \left\| x^a - x^i \right\|_2 = \left[\sum_{k=1}^{n} \left| x_k^a - x_k^i \right|^2 \right]^{\frac{1}{2}} = \sqrt[2]{\left| x_1^a - x_1^i \right|^2 + \left| x_2^a - x_2^i \right|^2 + \left| x_3^a - x_3^i \right|^2 + \left| x_4^a - x_4^i \right|^2}$$

$$d_2 = \left\| x^a - x^j \right\|_2 = \left[\sum_{i=1}^{n} \left| x_k^a - x_k^j \right|^2 \right]^{2} = \sqrt[2]{\left| x_1^a - x_1^j \right|^2 + \left| x_2^a - x_2^j \right|^2 + \left| x_3^a - x_3^j \right|^2 + \left| x_4^a - x_4^j \right|^2}$$

$$d_1 = \sqrt[2]{|5.6 - 4.9|^2 + |1.4 - 1.5|^2 + |6.1 - 6.3|^2 + |2.6 - 2.5|^2} = \sqrt[2]{0.49 + 0.01 + 0.04 + 0.01}$$

$$d_1 = \sqrt[2]{0.55}$$

$$d_2 = \sqrt[2]{|5.6 - 5.3|^2 + |1.4 - 1.9|^2 + |6.1 - 6.4|^2 + |2.6 - 2.7|^2} = \sqrt[2]{0.09 + 0.25 + 0.09 + 0.01}$$

$$d_2 = \sqrt[2]{0.44}$$

Since $d_2 < d_1$, the pattern x^a belongs to the class ϖ_3 which, as is known, is the correct class.

6 Conclusions and Future Work

By using Minkowski's metrics (the three particular chosen cases) in the k-NN classifier, we can see that even though the results shown by metric 1 are very similar to those given by the euclidean metric, the latter is more exact. We can also see that the error caused by the infinite metric is much greater than that caused by metric 1 or the euclidean metric. Therefore, we conclude that the best metric to use in a k-NN classifier applied to the Anderson's Iris Plant Database is the euclidean metric, and this conclusion agrees to the hypothesis proposed by Snapp and Venkatesh in [5]. As future work we suggest applying the methodology herein proposed to do a broader study on a diverse set of databases, from different fields of application, and show statistical results in order to corroborate (or reject) the theoretical hypothesis suggested in [5].

Acknowledgments

The authors of this paper wish to thank the CIC-IPN, COFAA-IPN, SA-IPN and CONACYT for their economical support. Also, we thank Itzama Lopez Yañez, Masters student at CIC-IPN for his help in style-checking this paper.

References

1. Cover, T. M. and Hart, P. E.: Nearest Neighbor Pattern Classification, IEEE Transactions on information theory, Vol. IT-13, 1 (1967) 21-27.
2. Goldstein, M.: kn-Nearest Neighbor Classification, IEEE Transactions on information theory, Vol. IT-18, 5 (1972) 627-630.
3. Short, R. D. and Fukunaga, K.: The Optimal Distance Measure for Nearest Neighbor Classification, IEEE Transactions on information theory, Vol. IT-27, 5 (1981) 622-627.
4. Waard (de), W. P.: An optimised distance method for character recognition, Pattern Recognition Letters, 16 (1995) 499-506.
5. Snapp, R. R. and Venkatesh, S. S.: k Nearest Neighbors in Search of a Metric, Proceedings of the IEEE International Symposium on Information Theory, Whistler, Canada (1995) 256.
6. Imiya, A.: A metric for spatial lines, Pattern Recognition Letters, 17 (1996) 1265-1269.
7. Jin, J. S. and Kurniawati, R.: Varying similarity metrics in visual information retrieval, Pattern Recognition Letters, 22 (2001) 583-592.
8. Peng, J., Heisterkamp, D. R. and Dai, H. K.: Adaptive Kernel Metric Nearest Neighbor Classification, Proceedings of IEEE International Conference on Pattern Recognition, Quebec City, Canada (2002) 33-36.
9. Duda, R. O. and Hart, P. E.: Pattern Classification and Scene Analysis, New York: John Wiley & Sons (1973).
10. Friedman, M. and Kandel, A.: Introduction to Pattern Recognition (Statistical, Structural, Neural and Fuzzy logic Approaches), Singapore, World Scientific (2000).

11. Sánchez-Garfias, F. A., Díaz de León, J. L. and Yáñez, C.: Reconocimiento automático de patrones. Conceptos básicos, IT 79, Serie Verde, CIC-IPN, México (2003).
12. Fisher, R. A.: The use of multiple measurements in taxonomic problems, Annals of Eugenics, v. 7 (1936) 179-188.
13. Hettich, S. & Blake, C.L. & Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998).