Neighborhood Counting Measure and Minimum Risk Metric

Hui Wang, *Member*, *IEEE*

Abstract—The *neighborhood counting measure* (NCM) is a similarity measure based on the counting of all common neighborhoods in a data space [5]. The *minimum risk metric* (MRM) [2] is a distance measure based on the minimization of the risk of misclassification. The paper by Argentini and Blanzieri [1] refutes a remark in [5] about the time complexity of MRM, and presents an experimental comparison of MRM and NCM. This paper is a response to the paper by Argentini and Blanzieri [1]. The original remark is clarified by a combination of theoretical analysis of different implementations of MRM and experimental comparison of MRM and NCM using straightforward implementations of the two measures.

Index Terms—Minimum risk metric, neighborhood counting measure, k-nearest neighbor.

1 Background

Neighborhood counting is a novel approach to measuring similarity between pairs of data items [5] which stipulates using the number of common neighborhoods of data items as a measure of their similarity. The notion of neighborhood can be defined for various types of data, and, for each type of data, this notion can be defined in different ways. Therefore, the neighborhood counting approach is generic.

Neighborhood counting was specialized to multivariate data, resulting in the *neighborhood counting measure* (NCM) [5]. NCM is presented in a simple formula, and it is easy to program and fast to compute. Experiments showed that, when used in k-nearest neighbor (kNN) classification tasks, NCM is competitive to some of the well-known distance/similarity measures, especially when k gets larger.

More recent studies have found that the neighborhood counting approach can also be applied to sequence data resulting in the *all common subsequences* similarity [6], [3], time series data [7], and tree structured data [4], resulting in the *all common embedded subtrees* similarity.

The *minimum risk metric* (MRM) [2] is a distance measure based on the minimization of the risk of misclassification. MRM is reviewed in the original paper [5], but is not compared against in the evaluation. Furthermore, there is a remark about MRM in [5]:

> *Our experiments showed that a straightforward implementation of MRM using the naive Bayes estimator is over 10 times more expensive computationally than a straightforward implementation of HEOM or DVDM.*

Argentini and Blanzieri [1] refuted this remark with some experimental evidence. It is shown that, if preprocessing is done, MRM is not so slow as remarked in [5]. It is further shown that MRM can be superior to NCM at times in kNN classification tasks.

This paper is intended to answer the following questions surrounding this debate, in order to present a fair, balanced view of MRM and NCM:

- Why was MRM not included in [5] as part of the evaluation?
- On what basis was the remark made in [5]?
- What is the relative performance of MRM and NCM, including both run time and classification accuracy?

For this, we will first of all summarize our response in the next section. We will then present an analysis of the two similarity measures from the time complexity point of view in Section 3, as well as experimental results on MRM and NCM in Section 4. Final remarks are presented in Section 5.

- *The author is with the School of Computing and Mathematics, University of Ulster, Jordanstown, Newtownabbey, Co. Antrim, BT37 0QB, Northern Ireland, UK. E-mail: h.wang@ulster.ac.uk.*

Manuscript received 21 Apr. 2009; accepted 7 June 2009; published online 11 Jan. 2010.
Recommended for acceptance by M. Figueiredo.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-04-0247.
Digital Object Identifier no. 10.1109/TPAMI.2010.16.

2 Summary of Our Response

MRM was not included in [5] as part of the evaluation for the following reasons:

- First, it is not possible to compare NCM with every similarity measure in a single paper.
- Second, MRM makes use of class separating information while NCM does not, so a direct comparison between the two may not be fair.

The remark was made on the basis of our "lazy" implementation of MRM, which does indeed take over 10 times as much time as NCM (see Section 3 for a detailed analysis and Section 4 for some experimental results).

The superior performance of MRM in classification, reported in [1], is not surprising as MRM takes into account class separating information while NCM does not. This comes at a price—consequently MRM cannot be applied if class information is not available, e.g., in (unsupervised) clustering. However, our own experiments (see Section 4) show that the two measures are at times comparable in terms of classification accuracy, although class separating information is taken into account in MRM but not in NCM.

Nevertheless, this comment paper [1] does fill a gap left by the original paper [5] in that a direct experimental comparison is done between MRM and NCM, and also that a remark in [5] is clarified.

3 Analysis of Time Complexity

3.1 Time Complexity of Computing NCM

Suppose the data space is defined by n attributes and a data set D consists of m data items. Let a_i ($i = 1, 2, \ldots, n$) be the domain of the ith attribute. A data item x is $< x_1, x_2, \ldots, x_n >$ where $x_i \in a_i$. It is assumed that all attributes have finite domains. For any two data items x and y, the NCM similarity between them is

$$NCM(x, y) = \prod_{i=1}^{n} c(x_i, y_i), \tag{1}$$

where $c(x_i, y_i)$ is given by

$$c(x_i, y_i) = \begin{cases} (\max(a_i) - \max(\{x_i, y_i\}) + 1) \times (\min(\{x_i, y_i\}) - \min(a_i) + 1), \\ \text{if } a_i \text{ is numerical} \\ 2^{m_i - 1}, \text{if } a_i \text{ is categorical and } x_i = y_i \\ 2^{m_i - 2}, \text{if } a_i \text{ is categorical and } x_i \neq y_i, \end{cases} \tag{2}$$

where $m_i = |a_i|$, $\max(a_i)$, and $\min(a_i)$ are the maximal and minimal elements in a_i if attribute i is numerical.

Given m and n, computing $c(\cdot)$ takes constant time $O(1)$. Therefore, computing $NCM(\cdot)$ takes $O(n)$ time, irrespective of m. Consequently, classifying a set of K data items takes time proportional to $K \times O(n)$.

TABLE 1
Five-Fold Cross Validation Results of k-Nearest Neighbor Classification Using MRM and NCM: Time and Classification Accuracy (CV5) over $k = 1$ and $k = 10$

			k=1				k=10			
			mrm		ncm		mrm		ncm	
Data	#Inst	#Attr	CV5	Time	CV5	Time	CV5	Time	CV5	Time
anneal	798	38	85.7	2858.4	76.2	100.9	85.7	2898.9	76.2	101.1
audiology	226	69	29.8	1749.6	74.2	0.9	45.7	1817.4	76.6	0.9
auto	205	26	57.6	181.2	76.1	4.0	69.3	180.6	75.1	4.0
breast-cancer	286	9	75.9	14.4	67.8	0.5	74.8	15.5	73.8	0.6
bridges-v1	108	11	53.1	20.3	56.5	0.3	61.9	20.7	66.7	0.3
bridges-v2	108	11	56.3	15.1	53.7	0.2	63.6	15.3	60.2	0.2
credit-rating	690	15	83.8	196.1	83.6	13.3	83.6	198.6	87.0	13.3
ecoli	336	8	86.6	369.0	73.5	3.6	86.6	372.3	80.0	3.6
german-credit	1000	20	58.6	539.5	71.5	28.3	65.4	547.0	74.3	28.4
glass	214	10	57.9	92.3	71.0	1.8	58.4	93.1	73.8	1.8
heart-statlog	270	13	84.4	24.7	78.9	2.7	84.4	24.4	80.4	2.8
hepatitis	155	19	85.8	5.9	79.4	0.7	85.8	5.8	81.9	0.7
horse-colic	368	27	78.0	76.4	74.2	9.5	78.0	76.0	83.7	9.6
ionsphere	351	34	90.3	104.6	90.9	14.8	90.3	104.5	90.6	14.9
iris	150	4	94.7	7.0	94.0	0.5	94.7	6.9	95.3	0.5
pima-diabetes	768	8	66.9	269.0	67.7	20.1	76.2	274.5	72.8	20.1
primary-tumor	339	17	39.4	1825.6	33.9	1.5	43.0	1810.9	38.7	1.5
sonar	208	60	81.3	57.6	85.6	9.4	81.3	57.7	87.1	9.4
soybean	307	35	83.7	3756.6	90.9	8.7	84.0	3827.3	90.9	8.8
tic-tac-toe	958	9	70.4	352.9	49.6	3.6	70.4	359.9	90.0	3.6
vehicle	846	18	57.7	1828.6	69.6	58.0	58.4	1835.0	72.0	58.2
vote	435	16	90.3	47.8	91.0	1.0	90.3	52.1	93.1	1.0
wine	178	13	98.3	25.1	94.4	1.9	98.3	25.0	95.5	1.9
yeast	1484	8	41.2	24708.7	53.2	73.1	49.9	25031.8	58.5	73.1
zoo	101	17	90.1	8.9	98.1	0.1	90.1	8.4	96.1	0.1
AVERAGE			75.3	1894.5	76.0	13.3	77.0	1917.4	80.8	13.3

3.2 Time Complexity of Computing MRM

The same notation is adopted here. For two data items x and y, the MRM distance is:

$$MRM(x,y) = r(x,y) = \sum_{i=1}^{C} p(c_i|x)(1 - p(c_i|y)), \quad (3)$$

where C is the number of classes and $p(c_i|x)(1 - p(c_i|y))$ is the risk of misclassifying x when it is in class c_i. It is clear that MRM depends on the availability of class separating information in $p(c_i|x)$.

Calculating MRM can be done in one of three ways:

- **Eager approach**: We construct a function from D as an estimate of $p()$. Every time we want to calculate $MRM(x,y)$, we only need to call function $p()$ without the need of going through D. In this approach, the time it takes to compute $MRM(x,y)$ is $O(C)$, where C is the number of classes, plus some one-off overhead for constructing $p()$. This one-off overhead is proportional to $m \times n$. Therefore, the time complexity for computing $MRM(x,y)$ is $O(C) + O(mn)$. Since the overhead is one-off, classifying a set of K data items takes time proportional to $K \times O(C) + O(mn)$.
- **Semi-eager approach**: We go through D to calculate some statistics needed to calculate $p()$, and store them somewhere. When we want to calculate $MRM(x,y)$ we employ these statistics. In this approach, the time it takes to calculate $MRM(x,y)$ is $O(C) + O(mn)$, where the second term is the one-off overhead for calculating and storing the statistics. Since the overhead is one-off, classifying a set of K data items takes time proportional to $K \times O(C) + O(mn)$.
- **Lazy approach**: We do not construct function $p()$ or store statistics. Instead, every time we want to calculate $MRM(x,y)$, we go through D to estimate both $p(c_i|x)$ and $p(c_i|y)$. The time for this is proportional to $O(mn)$. Therefore, the time to compute $MRM(x,y)$ is $O(C \times 2 \times mn) = O(Cmn)$. Consequently, classifying a set of K data items takes time proportional to $K \times O(Cmn)$.

It is clear that the lazy approach has a higher time complexity than the two eager approaches. However, the lazy approach does not need a preprocessing step, and it does not need additional space to store the probability values.

Andrea and Enrico remarked, "It is important to note that the only improvement in our implementation with respect to a straightforward implementation is that we compute the estimates $p(c_i|y)$ for y in the training set only once." This is a semi-eager approach as they need to compute and store the probability estimates in preprocessing.

4 Our Implementation of MRM and Experimental Results

We implemented MRM with a lazy approach when the paper [5] was prepared, where the naive Bayes method was used to estimate the probability terms. It turned out the MRM-based kNN was more than 10 times slower than the kNN using other distance measures. This is the observation behind the remark [5] "Our experiments showed that a straightforward implementation of MRM using the naive Bayes estimator is over 10 times more expensive computationally than a straightforward implementation of HEOM or DVDM." The phrase "straightforward implementation" was not explained further, but it actually meant the lazy approach.

4.1 Experimental Results

Experimental results based on our implementation of MRM are presented in Table 1. It can be seen that the runtime of MRM-based kNN is much higher than that of NCM-based kNN. It can also be seen that the two similarity measures are comparable on classification accuracy for this set of data. The source code of our implementation of MRM and NCM, and the data used in the experiment are available from the author.

5 Discussion and Conclusion

Based on the analysis in Section 3 and the experimental results in Section 4 we can conclude that

- MRM is indeed very slow in kNN classification tasks if a lazy approach is taken, although it can be speeded up by preprocessing, as shown in the comment paper [1].
- MRM makes use of class separating information, whereas NCM does not.
- It appears that NCM is comparable at times to MRM in terms of kNN classification accuracy, even though MRM makes use of class separating information. However, a systematic, independent evaluation is needed before any definite conclusion can be drawn with regard to their relative performance.

References

[1] A. Argentini and E. Blanzieri, "About Neighborhood Rounting Measure Metric and Minimum Risk Metric," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 4, pp. , Apr. 2010.
[2] E. Blanzieri and F. Ricci, "Probability Based Metrics for Nearest Neighbor Classification and Case-Based Reasoning," *Lecture Notes in Computer Science,* vol. 1650, pp. 14-29, 1999.
[3] C. Elzinga, S. Rahmann, and H. Wang, "Algorithms for Subsequence Combinatorics," *Theoretical Computer Science* vol. 409, no. 3, pp. 394-404, 2008.
[4] Z. Lin, H. Wang, S. McClean, and C. Liu, "All Common Embedded Subtrees for Measuring Tree Similarity," *Proc. Int'l Symp. Computational Intelligence and Design,* pp. 29-32, 2008.
[5] H. Wang, "Nearest Neighbors by Neighborhood Counting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 6, pp. 942-953, June 2006.
[6] H. Wang, "All Common Subsequences," *Proc. Int'l Joint Confs. Artificial Intelligence,* pp. 63-640, 2007.
[7] H. Wang and Z. Lin, "A Time Weighted Neighbourhood Counting Similarity for Time Series Analysis," *Proc. Int'l Conf. Rough Set and Knowledge Technology* 2008.