A flexible and robust similarity measure based on contextual probability

Hui Wang *
School of Computing and Mathematics
University of Ulster at Jordanstown
Northern Ireland
h.wang@ulster.ac.uk

Werner Dubitzky
School of Biomedical Science
University of Ulster at Coleraine
Northern Ireland
w.dubitzky@ulster.ac.uk

Abstract

Arguably, analogy is one of the most important aspects of intelligent reasoning. It has been hypothesized that, given suitable background knowledge, analogy can be viewed as a logical inference process. This study follows another school of thought that argues that similarity can provide a probabilistic basis for inference and analogy. Most *similarity measures* (which are frequently viewed as being conceptually equivalent to distance measures) are restricted to either nominal or ordinal attributes, and some are confined to classification tasks. This paper proposes a flexible similarity measure that is task-independent and applies to both nominal and ordinal data in a conceptually uniform way. The proposed similarity measure is derived from a probability function and corresponds to the intuition that if we consider all neighborhoods around a data point, the data points closer to this point should be included in more of these neighborhoods than more distant points. Experiments we have conducted to demonstrate the usefulness of this measure indicate that it fares very competitively with commonly used similarity measures.

1 Introduction

Natural and artificial intelligent agents rely on different inference and reasoning strategies to achieve their goals and maintain their intended behavior. Arguably, two of the more successful strategies are based on the notions of *analogy* [1] and *similarity*. This is evidenced by the large volume of AI research in the areas of *analogical reasoning* and *case-based reasoning* [Vosniadou and Ortony, 1989; Kolodner, 1993]. In the context of available background knowledge, analogy can be viewed as a logical inference process [Davies and Russell, 1987]. Another way of looking at analogy holds that similarity can provide a probabilistic basis for inference, and that a quantitative framework can be developed for the probability that an analogy is correct as a function of the degree of similarity measured or observed [Russell, 1986]. This study is concerned with the latter view of modeling analogy.

Distance measures (or, equivalently, similarity measures) are central to many areas related to AI, including reasoning under uncertainty, knowledge-based systems, machine learning, pattern recognition, data mining, analogical, case-based, instance-based reasoning, and to other fields such as statistics, operations research and decision theory. A large number of distance metrics and measures are available and a particular choice may have considerable implications for the success of the problem that is to be addressed.

Distance functions are broadly categorized into those that can handle *ordinal* input data, *nominal* input data, and *heterogeneous* input data, consisting of both ordinal and nominal data [2]. Ordinal distance functions make use of the intrinsic total ordering relation in the underlying attribute values, which are either continuous (e.g. weight of an object) or discrete (e.g. number of obstacles). A nominal or symbolic attribute is a discrete attribute whose values do not necessarily exhibit a total order relation. For example, an attribute representing the role of crew member may have the values *scientist*, *mission specialist* and *commander*. Using an ordinal distance measurement on such values is meaningless.

Because the notion of 'distance' is intrinsically numerical, most available distance measures are defined for data with ordinal attributes. However, distance measures that can

* Also LITA, Université de Metz, Ile du Saulcy, 57045 Metz Cedex, France.

[1] Here we consider the subject of analogy in the following, relatively narrow, sense: objects are represented by feature vectors rather than the typical graphs; we do not consider transfer of knowledge from one object to another; we do not consider cross-domain knowledge but assume all objects share common set of attributes.

[2] Here we consider two scales of measurement: ordinal and nominal. In an ordinal scale, values (discrete or continuous) are ordered whereas in a nominal scale, (discrete) values are unordered.

process nominal attributes are required by many modern AI algorithms. Some measures that handle nominal attributes do exist. The most well known measure of this kind is the Value Difference Metric (VDM) [Stanfill and Waltz, 1986]. It is defined in terms of attribute values that are conditioned by posterior probabilities of a class. Hence, like some other distance measures, the VDM is restricted to classification tasks.

To cope with heterogeneous data, containing both nominal and ordinal attributes, two approaches are commonly taken: (1) The data is transformed into one of the two data types that complies with the distance measure used, and (2) two types of distance measures are combined and handle the data separately in accordance with the data type (see, for example, [Wilson and Martinez, 1997]). However, both approaches are problematic when it comes to the interpretation of the results.

The novel distance measure proposed in this study is called *Neighborhood Counting Metric* (NCM). It is derived from a probability function and it can handle both nominal and ordinal attributes in a conceptually uniform way. Intuitively, it can be understood or interpreted as follows (see Figure 1): If we consider all neighborhoods, N, around a data point, t, then those data points closer to t should be included in more of these neighborhoods, N, than points that are not close to t. Usually neighborhoods are interpreted in terms of distance. However, if we adopted this interpretation, we would define one distance function in terms of another. To avoid this dilemma, we interpret neighborhood without distance through the concept of *hypertuples* [Wang *et al.*, 1999] for both nominal and ordinal attributes. As a result, our distance measure applies to both ordinal and nominal attributes in a uniform way.

The new NCM is conceptually simple, it is straightforward to implement, and it has the added property that it is independent of the underlying analytical or reasoning task (e.g. classification). The measure's clear and unambiguous meaning is defined by *the number of neighborhoods of a query that include or cover a given data point.*

Paper outline: Section 2 presents a short review of distance measures relevant to this study; it is followed by Sections 3 and 4, which present mathematical details of the new similarity measure. In Section 5 the empirical evaluation of the method is presented and its results discussed. The paper concludes with a summary and a brief discussion of future work.

2 A brief review of important distance functions

Two of the most common distance functions are *Euclidean distance* and the *Hamming distance*. The former is restricted to ordinal and the latter is usually used for nominal attributes. The *Heterogeneous Euclidean-Overlap Metric* (HEOM) [Wilson and Martinez, 1997] combines the Hamming and the normalized Euclidean distance functions and can therefore be used on heterogeneous data.

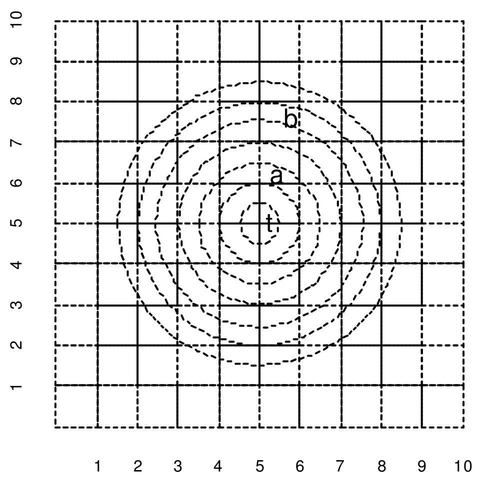

Figure 1: *An illustrative example. Each of the 7 concentric circles represents a neighborhood of data point t defined by some distance measure. Data point a is covered by 5 neighborhoods whereas b by only 2. Geometrically, a is clearly closer (more similar) to t than b.*

The *Value Difference Metric* (VDM) [Stanfill and Waltz, 1986] is designed to handle nominal attributes in classification tasks only. It uses pre-computed statistical properties from available training data and can be interpreted in terms of probabilities. The *Heterogeneous Value Difference Metric* (HVDM) [Wilson and Martinez, 1997] combines the Euclidean distance and the VDM. It is therefore able to handle heterogeneous data, but because of its VDM heritage it is confined to classification tasks.

The *Interpolated Value Difference Metric* (IVDM) [Wilson and Martinez, 1997] extends the VDM to scenarios involving ordinal attributes. In the learning phase, it employs a discretization step to collect statistics and determine the probability for the discretized values ($P_{a,x,c}$ in the VDM formula), but then retains the continuous values in the training data for later use in the application or testing phase. The IVDM requires a non-parametric probability density estimation to determine the probability values for each class. The *Discretized Value Difference Metric* (DVDM) is the same as the IVDM except that it avoids the retention of the original continuous values but uses only the discretized values.

[Blanzieri and Ricci, 1999] introduced the *Minimal Risk Metric* (MRM), a probability-based distance measure for classification and case-based reasoning. It minimizes the risk of misclassification and depends on probability estimation techniques. As a result, the MRM exhibits a high computational complexity and is task-dependent.

3 A probability function

Let V be a (non-empty) universe of discourse. A Σ-field $\mathcal{F}$ on V is a collection of subsets of V, such that:

1. $V \in \mathcal{F}$;
2. If $A \in \mathcal{F}$ then $A' \in \mathcal{F}$, where A' is the complement of A;
3. If $A, B, \cdots \in \mathcal{F}$, then $A \cup B \cup \cdots \in \mathcal{F}$.

A probability function P over V is a mapping from $\mathcal{F}$ to $[0,1]$ satisfying the three axioms of probability [Ash, 1972]. If P is restricted to V, it is called a probability mass function (discrete) or probability density function (continuous).

Suppose we have a probability function P as defined above. For $X \in \mathcal{F}$ let $f(X)$ be a non-negative measure of X satisfying $f(X_1 \cup X_2) = f(X_1) + f(X_2)$ if $X_1 \cap X_2 = \emptyset$. As an example, we can take $f(X)$ for the cardinality of X.

Consider a function $G : \mathcal{F} \to [0,1]$ such that, for $X \in \mathcal{F}$,

$$G(X) = \sum_{E \in \mathcal{F}} P(E) f(X \cap E)/K$$

where $K = \sum_{E \in \mathcal{F}} P(E) f(E)$. It can be easily shown that G is a probability function.

$G(X)$ is calculated from all those $E \in \mathcal{F}$ that overlap with X (i.e., $f(X \cap E) \neq 0$). These E's are relevant to X and serve as the *contexts* in which $G(X)$ is induced. Each such E is called a *neighborhood* of X. In other words a neighborhood of X is an element E of $\mathcal{F}$, i.e., a subset of V, such that E overlaps X.

In practice P is usually not known, but a *sample* of data drawn from the data space according to P is commonly available. P can be estimated from the sample by either parametric or non-parametric techniques. In some tasks (e.g., classification) the point-wise probability is needed. When there is insufficient data, which is not uncommon, the point-wise probability may be difficult to estimate using non-parametric methods. However, it is likely that probability can be estimated for some *regions* (contexts) in the data space based on the sample. Using the G probability function, we can estimate the G probability for any single data point from knowledge of the P probabilities of various regions or contexts. This provides us with a formal way of inference about probability under incomplete (hence uncertain) situations.

To calculate the G probability for $X \in \mathcal{F}$ in practice, we need to

1. Find the set of all neighborhoods appropriate to the problem at hand,
2. Estimate the P probability for every neighborhood,
3. Finally, estimate (calculate) the G probability for X according to its definition.

In classification tasks, we can further calculate the conditional G probability of a class given a single data point in a similar fashion. While this may appear to be computationally expensive, it is often possible to derive a simple formula for G.

Depending on what $\mathcal{F}$ is and how neighborhoods are defined, we can use the G probability for different tasks. In Section 4 we take $\mathcal{F}$ to be the set of all regions in a multi-dimensional space definable as hypertuples and our similarity measure is derived from this interpretation.

3.1 Estimation of G

This section describes how G is estimated for a data point from data samples. Let D be a sample of data drawn from V according to an unknown probability distribution P. Our aim is to calculate $\hat{G}(t|D)$ for any $t \in V$.

To calculate G we need P, which can be estimated from data assuming the *principle of indifference* as follows: For any $E \in \mathcal{F}$,

$$\hat{P}(E|D) = |E|/n$$

where $|E|$ is the number of elements in E and n is the number of elements in D. Additionally we assume that $f(X) = |X|$ for $X \in \mathcal{F}$.

We then have,

$$\begin{aligned}
\hat{G}(t|D) &= \sum_{E \in \mathcal{F}} \frac{\hat{P}(E|D) \times f(E \cap t)}{K} && \text{definition} \\
&= \sum_{E \in \mathcal{F}} \frac{\hat{P}(E|D) \times |E \cap t|}{K} && \text{specification of } f \\
&= \sum_{E \in \mathcal{F}, t \in E} \frac{\hat{P}(E|D)}{K} && \text{assumption that } t \text{ is in } E \text{ so } E \cap t = t \\
&= \frac{1}{nK} \sum_{E \in \mathcal{F}, t \in E} |E| && \text{estimation of } P \text{ by principle of indifference} \\
&= \frac{1}{nK} \sum_{x \in D} cov(t,x) && \text{expansion and then re-organisation}
\end{aligned}$$

where K is a normalisation factor, and $cov(t,x)$ is the number of such $E \in \mathcal{F}$ that covers both t and x. We call this number the *cover* of x with respect to t.

To obtain $cov(t,x)$, a straightforward approach would be to iterate over all $E \in \mathcal{F}$ and check if E covers both t and x. Clearly, such a process is undesirable because of its exponential complexity. In Section 4 we will present an efficient way of calculating $cov(t,x)$.

4 Measuring distance through counting

This section considers an interpretation of neighborhood and derives a formula for calculating $cov(t,x)$, which is then taken as a measure of similarity or distance.

4.1 Neighborhood

Pursuing a non-distance-based conceptualization of neighborhood, we interpret a *neighborhood* as a *hypertuple* [Wang *et al.*, 1999], and a *neighbor* as a data point (or tuple) covered by some neighborhood. So a neighborhood of a tuple is a hypertuple that covers the tuple.

Hypertuples

Let $R = \{a_1, a_2, \cdots, a_n\}$ be a set of attributes, and $dom(a)$ be the domain of attribute $a \in R$. Furthermore let $V \overset{\text{def}}{=} \prod_{i=1}^{n} dom(a_i)$ and $L \overset{\text{def}}{=} \prod_{i=1}^{n} 2^{dom(a_i)}$. V is the *data space* defined by R, and L an *extended data space*. A (given) *data set* is $D \subseteq V$ – a sample of V.

The attributes can be either *ordinal* or *nominal*. For simplicity, we assume the domain of any attribute is finite, but the results are not limited to finite domains.

If we write an element $t \in V$ by $\langle v_1, v_2, \cdots, v_n \rangle$ then $v_i \in dom(a_i)$. If we write $h \in L$ by $\langle s_1, s_2, \cdots, s_n \rangle$ then $s_i \in 2^{dom(a_i)}$ or $s_i \subseteq dom(a_i)$.

An element of L is called a *hypertuple*, and an element of V a *simple tuple*. The difference between the two is that a field within a simple tuple is a value while a field within a hypertuple is a set. If we interpret $v_i \in dom(a_i)$ as a singleton set $\{v_i\}$, then a simple tuple is a special hypertuple. Thus we can embed V into L, so $V \subseteq L$.

Consider two hypertuples h_1 and h_2, where $h_1 = \langle s_{11}, s_{12}, \cdots, s_{1n} \rangle$ and $h_2 = \langle s_{21}, s_{22}, \cdots, s_{2n} \rangle$. We say h_1 is *covered* by h_2 (or h_2 *covers* h_1), written $h_1 \leq h_2$, if for $i \in \{1, 2, \cdots, n\}$,

$$\begin{cases} \forall x \in s_{1i}, \min(s_{2i}) \leq x \leq \max(s_{2i}) & \text{if } a_i \text{ is ordinal} \\ s_{1i} \subseteq s_{2i} & \text{if } a_i \text{ is nominal} \end{cases}$$

Furthermore the *sum* of h_1 and h_2, written by $h_1 + h_2 \overset{\text{def}}{=} \langle s_1, s_2, \cdots, s_n \rangle$, is: for each $i \in \{1, 2, \cdots, n\}$,

$$s_i = \begin{cases} \{x \in dom(a_i) : & \min(s_{1i} \cup s_{2i}) \leq x \leq \max(s_{1i} \cup s_{2i})\}, \\ & \text{if } a_i \text{ is ordinal} \\ s_{1i} \cup s_{2i}, & \text{if } a_i \text{ is nominal} \end{cases}$$

The *product* operation $*$ can be similarly defined. It turns out that $\langle L, \leq, +, * \rangle$ is a lattice.

For a simple tuple t, $t(a_i)$ represents the projection of t onto attribute a_i. For a hypertuple h, $h(a_i)$ is similarly defined.

A hypertuple can be generated by taking one subset from each attribute. Let a_i be an attribute, $i = 1, 2, \cdots, n$; $\mathcal{S}_i$ be the set of all subsets of the domain of a_i; and $N''_i \overset{\text{def}}{=} |\mathcal{S}_i|$. Then a hypertuple h is an element of $\prod_i \mathcal{S}_i$. Therefore the number of all hypertuples is $\prod_i N''_i$.

If a_i is nominal, then $\mathcal{S}_i = 2^{dom(a_i)}$ and so $N''_i = 2^{m_i}$, where $m_i = |dom(a_i)|$. If a_i is ordinal, then an element $s \in \mathcal{S}_i$ corresponds to an interval $[\min(s), \max(s)]$. As a result, some elements of $\mathcal{S}_i$ may correspond to the same interval, and hence become equivalent. In general we have the following number of *distinctive* intervals for an ordinal attribute: $N''_i = \sum_{j=0}^{m_i-1}(m_i - j) = \sum_{j=1}^{m_i} j = \frac{m_i(m_i+1)}{2}$.

Neighborhoods as hypertuples

For any simple tuple t, a neighborhood of t is taken to be a hypertuple h such that $t \leq h$; and a neighbor of t with respect to h is $x \in V$ such that $x \leq h$. By this definition any simple tuple has a neighborhood. At least the maximal hypertuple in the extended data space is a neighborhood of any simple tuple since the maximal hypertuple covers all simple tuples.

For a query $t \in V$, not all hypertuples in $\prod_i \mathcal{S}_i$ are neighborhoods of t. For a hypertuple h to be a neighborhood of t we must have $t(a_i) \in h(a_i)$ for all i. Therefore, to generate a neighborhood of t, we can take an $s_i \in \mathcal{S}_i$ such that $t(a_i) \in s_i$ for all i, resulting in a hypertuple $\langle s_1, s_2, \cdots, s_n \rangle$. If a_i is nominal, the number of $s_i \in \mathcal{S}_i$ such that $t(a_i) \in s_i$ is $N'_i = \sum_{i=0}^{m_i-1} \binom{m_i-1}{i} = 2^{m_i-1}$ since s_i is any subset of $dom(a_i)$ that is the super set of $t(a_i)$. If a_i is ordinal, this number is $N'_i = (\max(a_i) - t(a_i) + 1) \times (t(a_i) - \min(a_i) + 1)$ since $(\max(a_i) - t(a_i) + 1)$ is the number of ordinal values above $t(a_i)$, and $(t(a_i) - \min(a_i) + 1)$ is such a number below $t(a_i)$. Any pair of values from the two parts respectively forms an interval.

To summarize, the number of neighborhoods of t is $\prod_i N'_i$, where

(1)
$$N'_i = \begin{cases} 2^{m_i-1}, & \text{if } a_i \text{ is nominal} \\ (\max(a_i) - t(a_i) + 1) \times (t(a_i) - \min(a_i) + 1), \\ & \text{if } a_i \text{ is ordinal.} \end{cases}$$

Cover of simple tuples

Under the above interpretation of neighborhood, we know exactly the number of all neighborhoods of a given simple tuple t. Here we present an efficient way of calculating $cov(t, x)$, the number of neighborhoods of t that cover x, which is needed in Section 3.1.

Consider two simple tuples $t = \langle t_1, t_2, \cdots, t_n \rangle$ and $x = \langle x_1, x_2, \cdots, x_n \rangle$. A neighborhood h of t covers t by definition, i.e., $t \leq h$. What we need to do is to check if h covers x as well. In other words, we want to find all hypertuples that cover both t and x.

Eq.1 specifies the number of all simple tuples that cover t only. We take a similar approach here by looking at each attribute and explore the number of subsets that can be used to generate a hypertuple covering both t and x. Multiplying these numbers across all attributes gives rise to the number we require.

Consider attribute a_i. If a_i is ordinal, then the number of intervals that can be used to generate a hypertuple covering both x_i and t_i is as follows:

$$N_i = (\max(a_i) - \max(\{x_i, t_i\}) + 1) \times (\min(\{x_i, t_i\}) - \min(a_i) + 1).$$

If a_i is nominal, the number of subsets for the same purpose is:

$$N_i = \begin{cases} 2^{m_i-1}, \text{if } x_i = t_i \\ 2^{m_i-2}, \text{otherwise} \end{cases}$$

Recall that $m_i = |dom(a_i)|$.

To summarize, the number of neighborhoods of t covering x is $cov(t,x) = \prod_i N_i$, where

(2)

$$N_i = \begin{cases} (\max(a_i) - \max(\{x_i,t_i\}) + 1) \times (\min(\{x_i,t_i\}) - \min(a_i) + 1), & \\ \qquad \text{if } a_i \text{ is ordinal} & \\ 2^{m_i-1}, & \text{if } a_i \text{ is nominal and } x_i = t_i \\ 2^{m_i-2}, & \text{if } a_i \text{ is nominal and } x_i \neq t_i \end{cases}$$

4.2 Use of cover as similarity measure

We use $cov(t,x)$ as a measure of similarity between t and x, which we call the *Neighborhood Counting Metric* or simply *NCM*. That is, for any two tuples x and y, the NCM between them is

$$NCM(x,y) = cov(x,y) = \prod_{i=1}^{n} N_i \tag{3}$$

where n is the number of attributes and N_i is given by Eq.2. It is clear that $NCM(x,y) \geq 0$, $NCM(x,x) \geq NCM(x,y)$ and $NCM(x,y) = NCM(y,x)$. Therefore the NCM is reflexive and symmetric, the properties generally required for a similarity measure [Osborne and Bridge, 1997].

This measure can be interpreted intuitively as follows. If we consider all neighborhoods around a data tuple, those tuples closer to this tuple should be included in more neighborhoods and those farther away should be included in fewer neighborhoods (see Figure 1). As a result, closer tuples should be assigned higher cover values.

In contrast to its usual interpretation in terms of distance, we interpret the notion of a neighborhood without distance through the concept of *hypertuples* for both nominal and ordinal attributes. As a consequence, our novel distance measure applies to both ordinal and nominal attributes in a conceptually uniform way.

Incidentally, the NCM intrinsically handles missing values in a fashion consistent with other measures. Recall, that the NCM is a product of all N_i's, where i is attribute index. For two data tuples, t and x, if there is a missing value in t or x for attribute i, then N_i is set to 1. As a result, this attribute does not contribute towards the NCM value.

5 Evaluation

The new Neighborhood Counting Metric is task-independent and can therefore be used for classification, clustering and other analytical tasks involving distances or similarities. We empirically evaluated the NCM in the context of a classification task, using the *k-nearest neighbor* (k-NN) classification algorithm with and without distance-based (neighbor) weighting [Baily and Jain, 1978]. The purpose of the evaluation is to compare the new measure with some of the commonly used distance measures in a setup involving *heterogeneous* data. The evaluation uses public benchmark data sets from UC Irvine Machine Learning Repository, which were selected with respect to their balance of ordinal and nominal attributes.

We implemented a k-NN algorithm with the novel NCM as well as the measures HEOM, HVDM, IVDM, and DVDM. The computational runtimes of the MRM turned out too excessive, so it was excluded from the study. In the experiments k was set to $1, 6, 11, 16, 21, 16, 31$ and to 'MaxK' (i.e., k = the number of data tuples in the training data). We adopted a 10-fold cross-validation procedure, and for each measure and each k value we ran the cross-validation 10 times and recorded the results for subsequent analysis.

Due to space limitations, we report only some of the results in detail. Table 1 shows the results when weighting was used and $k = MaxK$.

A statistical significance analysis was carried out using the Student t-test (two samples, assuming unequal variances) with $\alpha = 0.05$ (at a 95% confidence level). For each data set, each measure and each k value, we ran a 10-fold cross-validation using k-NN 10 times with random partitioning of data, resulting in a sample of 10 values. For a pair of samples, by two different measures, we have a total of 20 values. So the critical value is 2.1. We then calculate the 't' values. If $t \geq 2.1$ or $t \leq -2.1$ the two samples are significantly different (the classification rate of one sample is significantly higher or lower than that of the other). Notice that every value in Table 1 is the average of a sample of 10 values.

From Table 1 we can see that based on our experimental design the NCM achieved 17 out of 20 'significantly' higher classification success rates compared to the other methods (last row in table). Looking at the subtotals for 'nominal', 'ordinal' and 'mixture' categories (reflected by subtotals from top to bottom), the NCM achieves the largest margin over its competitors for the 'nominal' data sets, followed by 'mixture' and then 'ordinal'. This suggests that, when all data tuples are taken into consideration, the NCM is clearly superior under all circumstances.

The details of the results involving other values for k are not shown. In terms of k-changes, we observe that without weighting there was no significant difference between the used measures. When weighting was used, there was still little difference for small k values. The general trend for all measures was: as k got larger the classification success rate increased slightly but soon started to decline. The difference then showed up as the NCM displayed a much slower rate of decline and, after $k > 11$, the NCM consistently outperformed the other four measures (see $k = MaxK$ as discussed above). We can conclude that the NCM produces less variable or more robust results than the other four measures.

6 Conclusion

Starting from a probability function, we have developed a novel similarity or distance measure, the Neighborhood Counting Metric, which can be used with ordinal, nominal and heterogeneous attributes in a conceptually uniform way. This measure is defined without reference to any particular analytical task (e.g., classification). This means that the measure is potentially useful for many reasoning, inferential and reasoning tasks and systems modeling analogy or similarity. Because the NCM is based on a simple, easy-to-implement mathematical formulation and has a computational complexity in the same order as the Euclidean distance measure, it is a prime candidate for practical AI methods and tools.

Our empirical evaluation demonstrates that in a k-NN-based classification task, the measure significantly outperforms its competitors when distance-based neighbor weighting is used and when k is not too small, in particular when all data tuples are taken into consideration. As k gets larger, the NCM displayed a consistently superior performance over the reference methods. The difference is most significant when all data tuples in a training data set are considered. This implies that the NCM is less sensitive to k than the other measures considered in this study. Given an application based on the k-NN algorithm, if the optimal value for k is not known, we can simply consider all or a relatively large set of data tuples without a significantly compromising performance. Therefore, the performance of this measure is more predictable than the other methods investigated in this study.

In our experiments, all attributes were assumed to have equal weights. Future work will investigate how to determine the best attribute weights to achieve improved performance, as well as an application of the NCM to clustering tasks.

Data	DVDM	NCM	HEOM	IVDM	HVDM
Audiology (Standardized)	22.9	***66.3***	44.2	23.0	22.9
Bridges2 (Standardized)	44.3	***60.8***	52.9	44.3	44.3
Primary Tumor	25.6	***44.9***	26.0	25.6	25.6
Soybean (Large)	84.6	***86.7***	84.6	84.6	84.6
TTT	65.8	***68.0***	65.3	65.8	65.8
Vote	90.9	**91.0**	88.8	90.9	90.9
Yeast	32.3	***33.5***	32.0	33.0	32.0
Zoo	78.8	***95.0***	69.5	80.0	78.8
avg (nominal)	59.3	***70.5***	61.3	59.5	59.3
Diabetes	63.5	**63.6**	63.5	63.5	63.5
Ecoli	***61.1***	50.8	43.1	50.1	43.1
Glass	47.9	***61.1***	47.2	55.6	47.2
Iono	68.1	***80.5***	63.2	64.3	63.2
Iris	93.2	91.3	90.7	***94.3***	90.7
Pima	63.0	63.0	63.0	**63.1**	63.0
Sonar	74.3	***85.7***	73.1	71.6	73.1
Vehicle	46.3	***60.1***	42.3	48.2	42.3
Wine	94.7	96.7	94.6	***98.1***	94.6
avg (ordinal)	73.9	***77.3***	70.1	74.0	70.1
Anneal	75.3	75.3	75.3	75.3	75.3
Australian	85.5	***87.5***	82.8	85.3	82.8
Auto	49.1	***67.9***	46.4	48.2	46.4
Breast-Cancer	75.7	**76.0**	75.7	75.7	75.7
Bridges1	44.3	***62.2***	49.1	44.3	44.3
Credit Screening	60.8	***88.3***	82.0	64.7	59.5
German	68.7	**69.0**	68.7	68.7	68.7
Heart	79.2	**82.3**	80.6	81.7	72.0
Hepatitis	76.6	***81.7***	76.6	76.6	76.6
Hors-Colic	77.9	***84.7***	64.4	79.4	64.4
avg (heterogeneous)	69.3	***77.5***	70.2	70.0	66.6
Sig. count	1	17	0	2	0

Table 1: Average results (classification success rates) over 10 runs of the algorithms, where neighbor weighting was used and k=‘MaxK’. Best values are shown in bold, and statistically significant values in bold italic.

References

[Ash, 1972] R.B. Ash. *Real Analysis and Probability*. Academic Press, New York, 1972.

[Baily and Jain, 1978] T. Baily and A. K. Jain. A note on distance-weighted k-nearest neighbor rules. *IEEE Trans. Syst. Man Cyber.*, 8(4):311–313, 1978.

[Blanzieri and Ricci, 1999] Enrico Blanzieri and Francesco Ricci. Probability based metrics for nearest neighbor classification and case-based reasoning. *Lecture Notes in Computer Science*, 1650:14–29, 1999.

[Davies and Russell, 1987] Todd R. Davies and Stuart J. Russell. A logical approach to reasoning by analogy. In *Proc. of IJCAI87*, pages 264–270, Milan, Italy, 1987.

[Kolodner, 1993] Janet Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Osborne and Bridge, 1997] Hugh Osborne and Derek Bridge. Models of similarity for case-based reasoning. In *Procs. of the Interdisciplinary Workshop on Similarity and Categorisation*, pages 173–179, 1997.

[Russell, 1986] Stuart J. Russell. Quantitative analysis of analogy. In *Proc of AAAI86*, pages 284–288, Philadelphia, PA, 1986. Morgan Kaufmann.

[Stanfill and Waltz, 1986] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communication of ACM*, 29:1213–1229, 1986.

[Vosniadou and Ortony, 1989] Stella Vosniadou and Andrew Ortony, editors. *Similarity and Analogical Reasoning*. Cambridge University Press, New York, USA, 1989.

[Wang *et al.*, 1999] H. Wang, W. Dubitzky, I. Düntsch, and D. Bell. A lattice machine approach to automated case-base design: Marrying lazy and eager learning. In *Proc. IJCAI99*, pages 254–259, Stockholm, Sweden, 1999.

[Wilson and Martinez, 1997] D. Randal Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.