AI Communications 26 (2013) 261–279
DOI 10.3233/AIC-130565
IOS Press

Specific-class distance measures for nominal attributes

Khalil El Hindi

Computer Science Department, College of Computer and Information Sciences, King Saud University, Riaydh, Saudi Arabia
E-mails: khindi@ksu.edu.sa, kelhindi@gmail.com

Abstract. The classification accuracy of many machine learning methods depends upon their ability to accurately measure the similarity between different instances. Similarity is measured using a distance metric or measure. In this work, several novel distance measures for nominal values are proposed. These distance measures exploit the class of a training example against which a new instance is compared. The experiments, conducted using 50 benchmark data sets, indicate that the proposed functions are superior in many cases to the Value Difference Metric (VDM) that is widely used in instance based learning. Some of the proposed measures have proven to be less sensitive to missing values and noise in the training data sets and have maintained good classification accuracy in the presence of unknown and noisy attribute values. Like VDM, the proposed measures work only with labelled training data sets which makes them unsuitable for unsupervised learning methods.

Keywords: Machine learning, instance-based learning, kNN algorithm, lazy learners, distance functions, similarity metrics

1. Introduction

The accuracy of many machine learning methods depends on their ability to accurately measure the similarity between instances. These methods include Instance-Based Learning (IBL) [1], self-organizing maps [12], radial basis functions [6] and k-means clustering [13]. In this work, we propose new distance measures for nominal values and compare them with other measures in the context of IBL.

Instance-based learners [1], such as the k Nearest Neighbor algorithm (kNN), classify a new instance based on its similarity to other classified instances. The kNN algorithm searches the training data set for the k most similar instances and assigns the class that has the most votes as a predicted class for the new instance.

Similarity is measured using a function that calculates the distance between two instances. One of these instances is usually a new instance (or vector) with an unknown or unclassified class whilst the other is an instance with a known class taken from a training data set. Clearly, the classification accuracy of Instance-based learners is highly dependent upon the quality of its similarity function. It is relatively easy to exploit the ordering relationship between ordinal values to measure their similarity. It is much harder, however, to measure the similarity between nominal values because of the lack of an ordering relationship.

The distance function used influences the bias of a learning algorithm [24]. This bias is an extra piece of information, besides the training examples, that allows a classifier to generalize. Without bias, a classifier would be unable to generalize beyond the training data [14,24]. It has been shown that no single machine learning algorithm can consistently outperform another over all possible application domains [16]. Therefore, no one distance function provides optimal generalization accuracy when summed over all application domains [24]. However, it makes sense to favor a distance function which performs better in the application domains which are likely to occur over another that performs well on application domains which are less likely to occur.

This work proposes several measures for nominal values that make use of the classes of training instances against which a new unclassified instance is compared. Accordingly, these measures are suitable for supervised machine learning methods which assume that the classes of the training instances are given. Furthermore, they assume that a training example, in which one of the values occurs, is given. The experimental results obtained on 50 bench-mark data sets indicate that these measures are superior for many application domains to the VDM function [17], which fails to make

use of the class of the training instance against which the new unclassified instance is being compared, and especially so when dealing with missing and/or noisy data.

In Section 2, a number of widely used distance metrics and measures are reviewed. Section 3 proposes certain novel distance measures. Section 4 discusses the different methods for dealing with nominal and ordinal attributes. Section 5 assesses how well the different measures cope with unknown values. Section 6 assesses how well the different measures cope with the noise in the data sets and proposes another measure that is more tolerant of a certain type of noise.

2. Distance-based similarity functions

A number of distance functions have been proposed in the literature [2,4,11,17–21,24], some of which can manage ordinal or nominal attributes but not both while others, for instance HVDM (Heterogeneous VDM) and Heterogeneous Euclidean-Overlap Metric (HEOM) [24], can manage both. More recent work in this area such as [11,21] focuses on learning a distance metric that is more suitable for a given classification problem. This work, by way of contrast, seeks to address the fundamental problem of measuring the similarity (distance) between nominal attribute values and proposes some general purpose measures that are application independent.

The HVDM and HEOM functions are two general application-independent distance metrics that are related to the functions proposed in this work. HVDM and HEOM combine different metrics for nominal and ordinal attributes. They differ mainly in the way they handle nominal values. HEOM uses the Hamming distance between nominal values whilst HVDM uses the VDM [17]. The Hamming distance between two nominal values is 0 if two values are equal and 1 otherwise. This over-simplified approach fails to make use of additional information provided by nominal attribute values that can improve generalization [24].

On the other hand, HVDM [24] uses VDM [17] to measure the distance between nominal values, which is a more sophisticated technique. VDM works by considering two nominal attribute values similar if they have similar classification (i.e., they occur with the same classes). The following example, adopted from [24], helps illustrate the idea. When classifying some fruits and vegetables given some of their characteristics (attribute values) such as color and taste, the colors red and green are more similar than red and blue. This is because some fruits and vegetable can be colored either red or green, and VDM takes this similarity into account.

A normalized version of VDM is defined as follows (this is the best normalized version as reported by Wilson et al. [24] and is the one used in our comparative study).

$$\mathrm{vdm}_a(x_a, y_a) = \sqrt{\sum_{c=1}^{C} \bigl(p(c|x_a) - p(c|y_a)\bigr)^2}, \quad (1)$$

where

- x_a and y_a are two values of attribute a,
- $p(c|x_a)$ is the conditional probability that the class is c given that attribute a has value x_a.

HVDM also deals with numeric values and missing values. It is defined as

$$\mathrm{HVDM}(x, y) = \sqrt{\sum_{a=1}^{m} d_a^2(x_a, y_a)}, \quad (2)$$

where

- x and y are two vectors; typically one is a training instance and the other is a vector that needs to be classified,
- x_a and y_a are the values of attribute a in the vectors x and y, respectively,
- and m is the number of attributes.

The definition of d_a is given below:

$$d_a(x_a, y_a) = \begin{cases} 1 & \text{if } x_a \text{ or } y_a \text{ is unknown,} \\ \mathrm{diff}_a(x_a, y_a) & \text{if } a \text{ is ordinal,} \\ \mathrm{vdm}_a(x_a, y_a) & \text{if } a \text{ is nominal.} \end{cases} \quad (3)$$

It is worth noting that HVDM uses a conservative approach in dealing with missing attribute values. If the attribute value of the training instance or the new instance is unknown, it assumes that the distance is 1, which is the maximum distance between any two values.

diff_a is defined as follows:

$$\mathrm{diff}_a(x_a, y_a) = \frac{|x_a - y_a|}{4\sigma_a}, \quad (4)$$

where σ_a is the standard deviation of attribute a. HVDM uses σ_a for normalization because, assuming

a normal distribution, approximately 95% of data values falls within 2 standard deviations of the mean. This makes HVDM less sensitive to extreme outlying values than for instance HEOM, which uses for normalization purposes the difference between the maximum and the minimum attribute values.

A distance function has to satisfy the following properties in order to qualify as a metric; if it does not, it can be called a distance measure. The properties are

(1) Non-negativity: $d(x, y) \geqslant 0$.
(2) Distinguishability: $d(x, x) = 0$.
(3) Symmetry: $d(x, y) = d(y, x)$.
(4) Subadditivity or triangle inequality: $d(x, z) \leqslant d(x, y) + d(y, z)$.

3. Making use of the class of a training example

The VDM function fails to take into account the class of a training example against which the new instance (or vector) is compared. We believe that this class provides valuable information that can be exploited to accurately measure the distance between nominal values. The following example illustrates the point.

If the fruit and vegetable classification task discussed above in Section 2 is continued, but with categorization on the basis of the nominal attribute taste, and with only two values: *sweet* and *acidic*, the following problem arises. Some vegetables and fruits (i.e. classes) may have only one value. For example, lemons can only be *acidic*. However, many other instances of fruits and vegetables can be *sweet* and/or *acidic*, as a result of which the VDM distance between *sweet* and *acidic* can be close to zero. This can be misleading when measuring the distance between a new instance with a sweet taste and an instance in the training set of the lemon class. The distance between *sweet* and *acidic* in this case should be as large as possible in order to prevent the classifier from classifying the new instance as lemon.

This suggests that it may be useful to make use of the class of a training instance, $class_{\text{train}}$, against which the new instance is being compared. If $class_{\text{train}}$ does not occur (in the training set) with the attribute value of the new instance, the distance should be large. On the other hand, if the attribute value frequently occurs with instances of $class_{\text{train}}$, the distance should be small.

This can be achieved by using either the conditional probability of the class of the training instance, $class_{\text{train}}$, given the attribute value of the new instance, val_{new}, or the conditional probability of val_{new} given $class_{\text{train}}$. The former is defined as

$$p(class_{\text{train}} | val_{\text{new}}) = \frac{p(val_{\text{new}} \cap class_{\text{train}})}{p(val_{\text{new}})} \tag{5}$$

and the latter is defined as

$$p(val_{\text{new}} | class_{\text{train}}) = \frac{p(val_{\text{new}} \cap class_{\text{train}})}{p(class_{\text{train}})}. \tag{6}$$

Notice that the two formulas differ only in the denominator. All the proposed measures, introduced in the next subsections, were tested using these two formulas. However the second formula gave better average classification accuracy. The simple explanation for this is that the first formula unduly gives more weight to attributes with more values. Therefore we will only use the second formula to define most of the new measures. That is the probability of the attribute value given the class of a training instance. We will, however, use the first formula in Section 6 to define a measure that is more tolerant to noise in the class attribute values.

The second formula can be used to measure the distance between two values val_{new} and val_{train}. The next step is to propose several measure functions for nominal attributes that exploit this idea. We call these functions Specific-Class Distance Measures (SCDM).

3.1. The proposed measures

In this section, we describe two measures that exploit the class of the training instance against which a new instance is being compared.

3.1.1. Specific-Class Distance Measures (*SCDM*)

This function simply takes the absolute value of the difference between the conditional probability of the attribute value of a new instance, val_{new}, given the class of a training example, $class_{\text{train}}$, and the conditional probability of the attribute value of a training instance, val_{train}, given $class_{\text{train}}$.

$$\text{SCDM}(val_{\text{new}}, val_{\text{train}}) = \left| p\big(val_{\text{train}} | class_{\text{train}}\big) - p\big(val_{\text{new}} | class_{\text{train}}\big) \right|. \tag{7}$$

The thought process underlying this metric is simple; if the two attribute values are equally likely to occur given the class of the training instance ($class_{\text{train}}$), then the two values are similar and the distance should be small (the minimum value is 0). If, on the other hand, one value is less likely to occur given $class_{\text{train}}$ then the two values are not similar and the distance should be large (the maximum value is 1).

3.1.2. Inverted SCDM (ISCDM)

The above measure takes into account the probability of val_{new} given $class_{\text{train}}$ in a way that may not always be intuitive. If this probability is high, the distance should be small regardless of the probability of val_{train} given $class_{\text{train}}$, but, this is not the case if the probability of val_{train} given $class_{\text{train}}$ is small. What could be a better measure is one that depends solely on the probability of val_{new} given $class_{\text{train}}$.

If the probability of val_{new} given $class_{\text{train}}$ is 0, then that attribute value does not occur with that class in the training set; therefore the distance should be 1 (the maximum). On the other hand, if the probability of val_{new} given $class_{\text{train}}$ is high the distance should be small. The distance should be inversely proportional to the probability of val_{new} given $class_{\text{train}}$, the higher this probability, the smaller the distance. ISCDM (as defined below) achieves that

$$\begin{aligned}&\mathrm{ISCDM}(val_{\text{new}}, val_{\text{train}})\\&\quad = 1 - p(val_{\text{new}}|class_{\text{train}}). \qquad (8)\end{aligned}$$

This is also the definition of the probability of having another attribute value (other than val_{new}) given $class_{\text{train}}$.

It is important to note that this function does not use the attribute value of the training instance. Using information about only one value when measuring the distance between two values may sound counter intuitive, but this causes the measure to be less sensitive to missing or noisy attribute values in the training set. This issue is investigated further in Sections 5 and 6. Furthermore, the distance between a pair of values may vary depending on which class the training value is from.

It is also worth noting that this function fails to satisfy 2 of the properties required of metrics as set out above, namely, symmetry and distinguishability. Therefore ISCDM is considered to be a measure and not a metric. However, in practice, it gives good classification results as we shall see in the next section.

3.2. Experimental results

To compare the classification accuracy of distance measures described in the previous section with VDM and HEOM, the kNN algorithm, as implemented in Weka [25], has been used (with $k = 3$) on several data sets. We have renamed the measures so that they start with the letter H (for heterogeneous), to distinguish them from other variations that will be presented in the following sections. The proposed measures compute the distance between two vectors of values using Eq. (2). All measures deal with ordinal attributes in the same way as HVDM (using Eq. (4)). All measures, except HISCDM, deal with missing values in the same way as the HVDM; that is, if one of the values is missing, the function returns 1. HISCDM is the only exception. Since it does not rely on the attribute value of a training instance, it works as usual if that value is missing. It returns 1 only if the missing value is the attribute value of the new instance.

Table 1 shows the classification accuracy of the kNN algorithm implemented using each measure and tested on 29 data sets obtained from the UCI Repository for Machine Learning [3]. Each of these data sets contains at least one nominal attribute. Since the introduced measures deal with ordinal attributes using the same method as the HVDM, it is unnecessary to compare the new measures using data sets containing only ordinal attributes. Ten-fold cross validation was used in all experiments. The figures shown in the table are the averages of the 10 folds.

The table shows the results of each of the proposed measures when applied to each data set compared to the results of HEOM and HVDM. The table also shows the result of comparing HEOM and HVDM. The better results are highlighted in bold, and the significantly better results are highlighted in bold and underlined. A paired t-test with a confidence level of 95% was used to determine if each difference was statistically significant.

The average classification accuracies of HEOM, HVDM, HSCDM and HISCDM are 73.12%, 73.17%, 73.13% and 75.46%, respectively. The average classification accuracies of HEOM, HVDM and HSCDM are very close. However, the average classification accuracy of HISCDM, is higher by 2.34% than HEOM and by 2.29% than HVDM.

Each column in the table shows the result of comparing two different measures or metrics. The last two rows of each column show firstly, the number of data sets on which the measures or metrics achieved better results, and secondly, the number of significantly better results at a confidence level of 95%. HEOM achieved

Table 1
The average classification accuracy obtained using each of the proposed measures and HEOM and HVDM

Data set	HEOM vs HVDM		HEOM vs HSCDM		HEOM vs HISCDM		HVDM vs HSCDM		HVDM vs HISCDM	
	HEOM%	HVDM%	HEOM%	HSCDM%	HEOM%	HISCDM%	HVDM%	HSCDM%	HVDM%	HISCDM%
Sick	96.21	**96.92**	96.21	**96.50**	96.21	**96.50**	**96.92**	96.5	**96.92**	96.5
Lung-cancer	76.67	**80.00**	**76.67**	73.33	76.67	**80.00**	**80.00**	73.33	80	80
Labor	**83.00**	74.33	**83.00**	70.67	83.00	**93.33**	**74.33**	70.67	74.33	**93.33**
Hepatitis	**82.33**	79.67	82.33	**85.00**	82.33	82.33	79.67	**85**	79.67	**82.33**
Heart-h	**77.08**	72.99	**77.08**	75.26	77.08	**79.47**	72.99	**75.26**	72.99	**79.47**
Heart-c	79.87	**84.15**	79.87	**80.86**	79.87	**80.87**	**84.15**	80.86	**84.15**	80.87
Haberman	69.03	70.26	69.03	**71.62**	69.03	69.03	70.26	**71.62**	**70.26**	69.03
Flags	52.03	**58.79**	52.03	**52.58**	52.03	**58.66**	**58.79**	52.58	**58.79**	58.66
Cylinder-bands	**63.52**	60.00	**63.52**	61.67	**63.52**	61.30	60.00	**61.67**	60.00	**61.3**
Credit-g	**73.40**	70.70	**73.40**	70.40	73.40	**74.40**	**70.7**	70.4	70.7	**74.4**
Credit-a	**82.75**	80.72	**82.75**	80.14	**82.75**	82.17	**80.72**	80.14	80.72	**82.17**
Bridges_version2	**57.36**	48.18	**57.36**	54.73	57.36	**57.73**	48.18	**54.73**	48.18	**57.73**
Bridges_version1	**53.55**	50.91	53.55	**54.73**	53.55	**58.36**	50.91	**54.73**	50.91	**58.36**
Arrhythmia	**62.84**	62.18	**62.84**	62.18	62.84	**63.06**	62.18	62.18	62.18	**63.06**
Audiology	62.49	**71.84**	62.49	**73.14**	62.49	**74.90**	71.84	**73.14**	71.84	**74.9**
Colicorig	**64.13**	60.61	64.13	**76.58**	64.13	**70.11**	60.61	**76.58**	60.61	**70.11**
Colic	81.52	**83.96**	**81.52**	81.25	81.52	**81.80**	**83.96**	81.25	**83.96**	81.8
Autos	**37.95**	37.90	37.95	**39.43**	37.95	**40.83**	37.9	**39.43**	37.9	**40.83**
Car	80.19	**90.39**	**80.19**	79.56	80.19	**81.48**	**90.39**	79.56	**90.39**	81.48
Breast-cancer	**71.32**	67.87	71.32	**73.52**	**71.32**	69.61	67.87	**73.52**	67.87	**69.61**
Anneal	97.33	97.33	**97.33**	93.10	97.33	**97.44**	**97.33**	93.1	97.33	**97.44**
Vote	92.41	**95.39**	92.41	**93.07**	92.41	**93.54**	**95.39**	93.07	**95.39**	93.54
Trains	**70.00**	60.00	**70.00**	60.00	70.00	70.00	60.00	60.00	60.00	**70.00**
Vowel	63.33	**68.28**	63.33	**67.78**	63.33	**64.14**	**68.28**	67.78	**68.28**	64.14
Primary-tumor	**43.64**	38.89	**43.64**	39.80	43.64	**44.22**	38.89	**39.8**	38.89	**44.22**
Nursery	79.29	**88.12**	79.29	**89.65**	79.29	**86.58**	88.12	**89.65**	**88.12**	86.58
Mushroom	**99.94**	99.93	**99.94**	99.35	**99.94**	99.93	**99.93**	99.35	99.93	99.93
Lymph	**83.76**	82.48	**83.76**	81.71	83.76	**86.38**	**82.48**	81.71	82.48	**86.38**
Soybean	83.45	**89.14**	**83.45**	83.28	83.45	**90.01**	**89.14**	83.28	89.14	**90.01**
Average	**73.12**	**73.17**	**73.12**	**73.13**	**73.12**	**75.46**	**73.17**	**73.13**	**73.17**	**75.46**
# Better	16	11	16	13	4	22	15	12	9	19
# Sig Better	5	10	4	4	0	8	6	2	3	8

better results than HVDM on 16 data sets (highlighted in the table in bold) of which only 5 were significantly better (highlighted in the table in bold and underlined). On the other hand, HVDM achieved better results than HEOM on 11 data sets, 10 of which were significantly better. HEOM achieved better results than HSCDM on 16 data sets, only 4 of which were significantly better. Finally, HSCDM achieved better results than HEOM on 13 data sets, 4 of which were significantly better.

HISCDM emerged as the winner compared with HEOM, achieving better results on 22 data sets, 8 of which were significantly better. HEOM achieved better results than HISCDM on only 4 data sets, none of which were significantly better. The table also shows the results of comparing the newly proposed measures with HVDM. When compared with HVDM, HSCDM achieved better results on 12 data sets, only 2 of which were significantly better, while HVDM achieved better results on 15 data sets, 6 of which were significantly better. Once again HISCDM emerged as the winner, achieving better results than the HVDM function on 18 data sets, 8 of which were significantly better, while HVDM achieved better results on only 9 data sets, with only 3 being significantly better.

It is worth stressing that, since the distance measure used affects the inductive bias of the algorithm, no one

measure can be consistently better than all other measures across all data sets. Moreover, having many measures with different inductive biases provides a pool of measures out of which one can select the most suitable measure for a given application domain. Table 1 shows that there are only 8 data sets on which no one measure achieves significantly better results. This is an indication that the introduced measures together with HEOM and HVDM form a sufficiently diverse pool of distance measures.

4. Dealing with nominal and ordinal attributes cohesively

In the previous section, we used heterogeneous measures for dealing with ordinal and nominal attributes. The inherent danger with this method is that a function may give undue weight to one or other of these types of attributes [24]. To avoid this problem, a cohesive method, capable of taking both types equally into account, is required. Two approaches can be used: discretization and interpolation. Since the HVDM and HISCDM emerged as the best two measures in the previous section, the scope of this section is limited to considering variations of these two measures only.

4.1. Discretization

An ordinal attribute can be discretized into a number of discrete values that can be dealt with as nominal attribute values. Although ordinal information is lost, discretization improves the average classification accuracy as we will see in the experimental results section. Discretization methods can be either supervised or unsupervised [7]. Supervised methods make use of class attribute values [5,10], while unsupervised methods rely on clustering techniques that do not make use of these values. In this work, we use Fayyad et al.'s [10] method as implemented in Weka [25]. The method uses the entropy function as a criterion to decide on the splitting point as used in the C4.5 algorithm [15]. It calls itself recursively to further split each interval into subintervals. The method uses Minimum Description Length as a stopping criterion.

4.2. Interpolation

One drawback of discretization is that two values at the opposite ends of a range (interval) might be considered equal. Moreover, two values that are close to each other, but that happen to fall in different intervals, are considered totally different [24]. To deal with such a problem, Wilson et al. [24] introduced a method based on linear interpolation. The method is called Interpolation (IVDM). In this section we review IVDM, and show how a similar method can be used to interpolate ISCDM.

IVDM uses VDM to compute the distance between, both nominal and ordinal values. IVDM uses discretization only to sample the probabilities $p(class|val)$ at the midpoints of each discretized range.

During classification, linear interpolation is used to approximate the probabilities $p(class|val)$ for two ordinal values. The approximated values of $p(class|val)$ are then used to compute the VDM value for a pair of ordinal values.

Another slightly modified method called the WVDM [24] attempts to find better sampling points than the midpoints. The WVDM is computationally more expensive than IVDM and the empirical results reported in [24] show little, if any, improvements over IVDM. Therefore, WVDM was excluded from our empirical comparisons in this work.

4.2.1. Interpolated ISCDM (IISCDM)

In a similar way to that used in IVDM, we used linear interpolation to develop an interpolated version of ISCDM. We used discretization to sample the probabilities $p(val|class)$ and linear interpolation during classification to approximate $p(val_{\mathrm{new}}|class_{\mathrm{train}})$ where val_{new} is an ordinal value. It is worth noting that this step is less computationally demanding than computing the IVDM value for a pair of ordinal values during classification. Computing $\mathrm{IVDM}(val_1, val_2)$ requires computing

$$\sqrt{\sum_{c=1}^{C}\left(p(c|val_{\mathrm{new}}) - p(c|val_{\mathrm{train}})\right)^2} \tag{9}$$

which takes time proportional to the number of classes, C, whilst computing ISCDM takes a constant time.

This added overhead on classification by the IVDM slows the classification process making it $\mathrm{O}(CN)$ instead of $\mathrm{O}(N)$, where C is the number of classes and N the number of instances in a training set. This could require significantly longer classification time for application domains with many classes such as character recognition. The interpolated ISCDM, on the other hand, does not add extra overhead to the classification process.

4.3. Experimental results

Once again, kNN was used, with $k = 3$, to compare the various cohesive methods. Table 2 summarizes the results of our experiments conducted on 38 data sets. Each figure in the table is the average obtained using 10-fold cross validation. The best results are shown highlighted in bold. There are more data sets in this section because in the previous section data sets that do not contain any nominal attributes were excluded, whereas in this section we excluded only data

Table 2

The average classification accuracy of heterogeneous and cohesive methods

Data set	HVDM	HISCDM	IVDM	IISDM	DISCDM	DVDM
Anneal	97.33	97.44	96.99	95.43	**99.00**	97.77
Arrhythmia	62.17	63.06	66.37	46.90	73.23	**74.34**
Arrhythmia	62.17	63.06	66.37	46.90	73.23	**74.34**
Audiology	71.84	**74.90**	71.68	74.78	74.78	71.68
Autos	37.90	40.83	33.17	40.49	**63.90**	58.54
Breast-cancer	67.87	**69.61**	67.83	69.58	69.58	67.83
Breast-w	95.85	95.85	95.71	92.27	96.71	**97.14**
Bridges_version1	50.91	58.36	52.34	**63.55**	**63.55**	54.21
Bridges_version2	48.18	57.73	48.60	**57.94**	**57.94**	48.60
Car	**90.39**	81.48	**90.39**	81.48	81.48	**90.39**
Colic	83.96	81.80	84.24	77.99	80.43	**85.05**
Colic.orig	60.60	70.11	64.95	70.11	**76.36**	65.22
Credita	80.72	**82.17**	80.58	78.70	82.03	80.87
Credit-g	70.70	**74.40**	69.80	73.10	73.40	72.30
Cylinder-bands	60.00	61.30	60.56	64.26	**67.22**	59.07
Diabetes	**74.48**	**74.48**	72.27	73.96	73.70	73.83
Ecoli	75.60	75.60	**77.68**	74.11	76.19	76.49
Flags	**58.79**	58.66	58.25	57.73	57.73	58.25
Glass	67.29	67.29	61.21	48.60	72.43	**77.10**
Haberman	70.26	69.03	70.59	72.22	**74.84**	74.51
Heart-c	84.15	80.87	85.15	81.85	84.16	**85.48**
Heart-h	72.99	79.47	73.13	76.19	**83.67**	79.25
Hepatitis	79.67	82.33	85.16	85.16	**90.32**	84.52
Labor	74.33	**93.33**	84.21	87.72	87.72	82.46
Liver-disorders	**57.97**	**57.97**	47.83	49.57	54.78	54.78
Lymph	82.48	86.38	84.46	83.11	**87.16**	81.76
Nursery	**88.12**	86.58	**88.12**	86.58	86.58	**88.12**
Optdigits	97.30	97.30	96.69	85.32	95.75	**97.38**
Pendigits	**99.32**	**99.32**	99.00	74.69	96.88	98.35
Primary-tumor	38.89	44.22	38.94	**44.25**	**44.25**	38.94
Segment	95.41	95.41	94.11	81.34	94.81	**95.80**
Sick	96.92	96.50	96.63	76.75	97.16	**97.43**
Sonar	48.56	48.56	52.88	58.65	**80.77**	69.23
Soybean	89.17	90.01	89.17	**90.04**	**90.04**	89.17
Trains	60.00	**70.00**	60.00	**70.00**	**70.00**	60.00
Vehicle	70.92	70.92	62.65	52.13	**72.22**	70.80
Vote	**95.40**	93.54	**95.40**	93.56	93.56	**95.40**
Vowel	**68.28**	64.14	62.32	42.42	62.83	62.22
Wine	93.82	93.82	93.26	93.26	97.75	**98.31**
Average	**74.17**	**75.91**	**74.01**	**71.73**	**78.55**	**76.65**
# Best	**8**	**9**	**4**	**5**	**15**	**12**

sets which did not contain ordinal attributes. The last row in the table shows the number of data sets on which the corresponding measure gave the best results.

We renamed the measures by prefixing each name with either an "I" (for Interpolation) or a "D" (for Discretization). For example, DISCDM refers to Discretized-ISCDM, where discretization was performed using Fayyad et al.'s method [10], while IISCDM refers to Interpolated-ISCDM. The table shows that the average classification accuracy obtained using discretization is higher than that obtained using heterogeneous and interpolation methods. This is true for both DVDM and DISCDM with average classification accuracies of 76.65% and 78.55%, respectively. DISCDM gave the best results (shown highlighted in bold), compared to all other measures, on 15 data sets, whilst DVDM gave the best results, compared to all other measures, on 12 data sets.

The heterogeneous methods (HVDM and HISCDM) achieved the second best results with average classification accuracy of 74.17% and 75.91% for HVDM and HISCDM, respectively. HISCDM achieved the best results on 9 data sets, while HVDM achieved the best results on 8 data sets.

The interpolation method achieved the worst results with average classification accuracies of 74.01% and 71.73% for IVDM and IISCDM, respectively. IVDM achieved better results than IISCDM, which could indicate that IISCDM might be more sensitive to the selected number of discrete values. We used 5 values in our experiments as suggested in [24]. IISCDM achieved the best results on 5 data sets, while IVDM achieved the best results on 4 data sets.

Our results, contrary to [24], show that discretization gives better results than interpolation. This is probably because we used a more sophisticated discretization method [10] than the simple discretization method used in [24]. In [24], each ordinal attribute was discretized into a fixed number of discrete values. This number is either five or the same as the number of class values if this number is greater than five.

5. Dealing with missing values

In the previous section, DISCDM and DVDM gave the best and second best results respectively. However since DISCDM does not use, during classification, the attribute value of the training examples, it may be better able to handle missing values in a training set than DVDM.

To study the effectiveness of DISCDM compared to DVDM in dealing with missing values, we used 50 data sets and deliberately flagged some attribute values in the training data as "missing", leaving the test sets untouched. We repeated the experiments with 0%, 5%, 10%, 15%, 20%, 25% and 30% missing values. Insertion of the missing values in the training sets was achieved by randomly selecting a number of attribute values and replacing these with blank values. Due to the random nature of the selection step, every experiment was repeated 5 times for each rate of missing values, other than 0%. Table 3 shows the average of the 5 experiments. We performed 10-fold cross validation for each data set and each ratio of missing values. The figures shown in the table for 0% missing values are the average results of the 10-folds cross validations, while each figure shown for all other missing-value rates is the average of 50 experiments (10-fold cross validation repeated 5 times). The last two rows in the table show the number of data sets on which the measures achieved better accuracy and significantly better accuracy with a 95% confidence level. Again, the better results are highlighted in bold in the table and the significantly better results are highlighted in bold and underlined.

At 0% missing values, DISCDM gave better results than DVDM on 24 data sets, of which 11 were significantly better at 95% confidence level. On the other hand DVDM gave better results than DISCDM on 22 data sets of which 10 were significantly better. The two measures achieved the same average accuracy on 4 data sets. At a 90% confidence level, DISCDM and DVDM gave significantly better results on 14 and 10 data sets, respectively. At an 80% confidence level, DISCDM gave significantly better results on 17 data sets, while DVDM gave significantly better results on only 12 data sets.

As the ratio of missing values increased, DISCDM became increasingly more accurate than DVDM, both in terms of average accuracy and number of data sets. This reflects a rapid degradation in the average accuracy of DVDM as the ratio of missing values increased, with average accuracy dropping from 80.61% at 0% missing values to 67.70% at 30% missing values, a decrease of 12.91%. Conversely, the average accuracy of DISCDM degraded at a much slower rate from 81.92% at 0% missing values to 80.13% at 30% missing values, a decrease of only 1.79%. Figure 1 illustrates the gap, in terms of average accuracy between the two measures, widening as the ratio of missing values increases.

Table 3

The average classification accuracy of DISCDM and DVDM at different rates of missing values in the training sets

Data set	0%		5%		10%		15%		20%		25%		30%	
	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM
Anneal	97.77	**99**	97.33	**98.76**	93.45	**98.96**	89.05	**98.6**	85.79	**98.98**	79.13	**99**	78.83	**98.69**
Anneal.ORIG	97.10	**97.66**	93.97	**97.46**	91.31	**97.66**	86.51	**97.33**	87.65	**96.57**	78.73	**94.95**	76.63	**91.00**
Arrhythmia	**74.34**	73.25	59.58	**72.08**	57.53	**73.24**	55.94	**73.99**	55.5	**73.68**	53.81	**73.63**	54.57	**75.13**
Audiology	71.84	**74.9**	36.95	**77.02**	27.63	**76.14**	27.75	**74.28**	27.15	**73.76**	25.46	**72.88**	26.08	**72.6**
Autos	58.57	**63.86**	58.49	**61.5**	57.17	**61.3**	50.12	**59.68**	47.45	**60.38**	39.91	**59.85**	36.29	**57.5**
Breast-cancer	67.87	**69.61**	67.78	**68.7**	66.46	**70.47**	68.76	**70.24**	68.01	**70.15**	65.78	**69.56**	66.39	**70.1**
Breast-w	**97.14**	96.71	**97.00**	95.37	**97**	95.79	**96.57**	95.82	**96.83**	95.8	**96.8**	96.63	**96.71**	96.34
Bridges ver1	53.45	**63**	57.93	**62.04**	54.35	**61.55**	53.78	**62.33**	50.09	**59.35**	48.55	**60.38**	47.87	**61.2**
Bridges ver2	48.18	**57.73**	47.67	**59.25**	47.25	**57.2**	44.2	**57.33**	47.55	**58.62**	44.75	**56.67**	46.16	**55.95**
Car	**90.39**	81.48	**89.1**	87.69	**88.21**	86.72	**86.86**	85.28	**86.24**	83.43	**84.59**	83.39	**82.71**	80.59
Colic	**85.05**	80.42	**82.93**	81.88	82.23	**82.81**	79.66	**82.43**	76.91	**82.49**	75.55	**82.54**	71.63	**82.7**
Colic.orig	65.2	**76.38**	71.02	**72.66**	69.17	**71.23**	70.58	**72.22**	71.01	**71.52**	70.28	**72.67**	70.32	**72.71**
Credit-a	80.87	**82.03**	**82.43**	81.07	**82.58**	81.94	**83.22**	82.43	**83.19**	82.55	81.91	**82.99**	79.04	**83.48**
Credit-g	72.3	**73.4**	**70.16**	69.96	**70.1**	70.94	70.38	**71.06**	66.34	**70.74**	66.4	**71.5**	65.82	**70.88**
Cylinder bands	59.07	**67.22**	60.3	**65.37**	57.56	**64.52**	59.93	**65.96**	56.44	**64.85**	56.7	**65.26**	58.7	**63.63**
Dermatology	**96.99**	96.72	95.84	**96.68**	93.44	**97.27**	90.83	**96.85**	79.78	**82.59**	86.14	**96.30**	86.54	**96.52**
Diabetes	**73.83**	73.69	73.62	**74.32**	**75.16**	74.22	74.63	**74.74**	74.42	**74.47**	73.67	**75.49**	72.28	**75.34**
Ecoli	**76.29**	75.98	**76.87**	76.02	76.03	**77.15**	72.32	**80.29**	71.27	**78.53**	67.39	**78.16**	70.95	**78.51**
Flags	**58.26**	57.61	50.06	**58.61**	49.41	**58.79**	44.13	**58.92**	41.77	**60.66**	39.31	**59.12**	37.39	**58.09**
Glass	**77.23**	72.51	73.94	**76.48**	72.46	**75.84**	69.26	**75.28**	63.41	**74.63**	60.91	**73.02**	60.97	**73.9**
Haberman	74.53	**74.85**	73.93	**74.26**	72.45	**73.09**	72.24	**72.96**	72.56	**72.97**	71.6	**73.34**	70.29	**71.85**
Heart-c	**85.45**	84.14	81.18	**84.14**	81.4	**83.68**	82.11	**83.42**	81.25	**84.1**	79.28	**83.83**	78.82	**83.63**
Heart-h	79.16	**83.57**	77.7	**81.54**	78.88	**81.21**	80.06	**81.14**	78.18	**80.74**	80.65	**80.79**	75.28	**81.74**
Heart-statlog	82.59	82.59	80.15	**82.59**	80.00	**81.85**	80.22	**82.22**	94.82	**98.10**	78.07	**81.93**	78.15	**82.96**
Hepatitis	84.38	**90.29**	82.64	**85.97**	82.72	**86.36**	81.97	**86.77**	79.09	**86.22**	78.98	**85.73**	72.73	**86.62**
Hypothyroid	**98.73**	98.49	98.14	**98.37**	97.45	**98.62**	96.09	**98.19**	89.18	**90.65**	93.73	**98.23**	93.22	**98.24**

Table 3

(Continued)

Data set	0%		5%		10%		15%		20%		25%		30%	
	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM
Lonosphere	89.46	**90.88**	89.80	**91.34**	89.74	**91.17**	89.24	**91.11**	69.59	**87.69**	89.29	**90.94**	87.93	**91.05**
Labor	83	**88**	85	**92.27**	86.87	**90.53**	86.53	**92.47**	86.93	**90.73**	88.87	**91.4**	84.27	**92.13**
Letter	**93.79**	91.13	**90.56**	90.42	85.08	**89.45**	78.33	**88.75**	**97.72**	96.91	57.70	**86.69**	41.80	**85.59**
Liver-disorders	54.84	54.84	55.92	**56.78**	**56.12**	55.85	**55.25**	55.2	**57.45**	54.32	55.83	**56.97**	**55.88**	54.25
Lung-cancer	80	80	**74**	73.33	72.67	**76.67**	70.17	**72.67**	70	**74.67**	71.33	**72**	71.33	**74**
Lymph	81.81	**87.1**	79.48	**80.94**	76.07	**81.11**	73.69	**81.37**	70.17	**79.75**	65.72	**80.43**	61.33	**81.5**
Mushroom	99.93	99.93	99.88	**100**	99.89	**99.98**	99.71	**99.96**	99.31	**99.95**	99.31	**99.88**	99.19	**99.87**
Nursery	**88.12**	86.58	88.03	**90.23**	87.21	**89.44**	86.74	**88.41**	85.66	**87.66**	85.06	**86.52**	83.9	**84.98**
Optdigits	**97.38**	95.75	93.33	**94.8**	84.46	**94.72**	69.01	**94.5**	52.44	**94.35**	50.89	**94.12**	42.2	**93.88**
Pendigits	**98.35**	96.88	**97.9**	97.17	**96.89**	96.52	95.27	**95.7**	92.33	**95.19**	88.32	**94.57**	84.21	**93.81**
Primary tumor	38.89	**44.22**	39.4	**42.47**	34.01	**42.22**	28.53	**42.82**	21.64	**42.58**	18.16	**43.18**	17.47	**43.53**
Segment	**95.8**	94.81	94.49	**95.65**	93	**95.37**	90.94	**94.58**	90.31	**94.88**	88.23	**94.29**	87.54	**94.07**
Sick	**97.43**	97.16	**96.97**	**96.98**	96.64	**96.97**	95.07	**96.85**	94.24	**96.83**	93.93	**96.63**	94.09	**96.68**
Solar-flare_1	**97.52**	97.21	**97.53**	97.16	**97.83**	97.52	**97.84**	97.47	**99.53**	98.91	**97.84**	96.42	**97.84**	96.84
Solar-flare_2	**99.53**	99.06	**99.53**	99.15	**99.53**	98.97	**99.53**	98.99	**99.53**	98.91	**99.53**	98.91	**99.53**	98.93
Sonar	69.26	**80.74**	71.39	**73.04**	66.61	**74.01**	61.56	**74.28**	64.47	**75.84**	55.33	**75.05**	51.61	**76.24**
Soybean	89.14	**90.01**	79.38	**89.25**	57.72	**88.58**	50.93	**89.05**	46.25	**88.25**	39.46	**87.81**	31.39	**87.49**
Spambase	89.31	**91.24**	86.83	**91.26**	82.57	**91.22**	81.71	**91.22**	77.28	**91.26**	75.94	**90.70**	74.34	**90.90**
Splice	**93.57**	86.87	86.42	**87.12**	67.62	**86.80**	62.13	**87.05**	60.50	**86.68**	55.18	**86.60**	51.47	**86.32**
Trains	60	**70**	**68**	62	56	**62**	**58**	50	**52**	50	56	**62**	50	**54**
Vehicle	70.8	**72.22**	67.87	**69.73**	63.54	**69.64**	60.02	**68.95**	56.67	**68.59**	55.26	**68.58**	53.5	**67.49**
Vote	**95.39**	93.54	**94.93**	93.96	**94.43**	93.5	**93.84**	93.36	**92.96**	92.77	**92.6**	91.99	92.05	**92.54**
Vowel	62.22	**62.83**	57.49	**57.98**	52.44	**57.58**	47.17	**55.6**	41.94	**57.27**	33.52	**55.05**	25.9	**53.56**
Wine	**98.33**	97.75	96.89	**97.71**	96.11	**97.36**	96	**97.84**	95.44	**96.93**	95.22	**97.61**	91.98	**96.94**
Average	**80.61**	**81.92**	**78.55**	**81.21**	**75.89**	**81.11**	**73.89**	**80.76**	**72.12**	**80.55**	**69.73**	**80.52**	**67.70**	**80.13**
# Better	**22**	**24**	**13**	**37**	**10**	**40**	**8**	**42**	**8**	**42**	**6**	**45**	**5**	**45**
# Sig. better	**10**	**11**	**5**	**24**	**5**	**32**	**3**	**35**	**5**	**37**	**2**	**37**	**3**	**42**

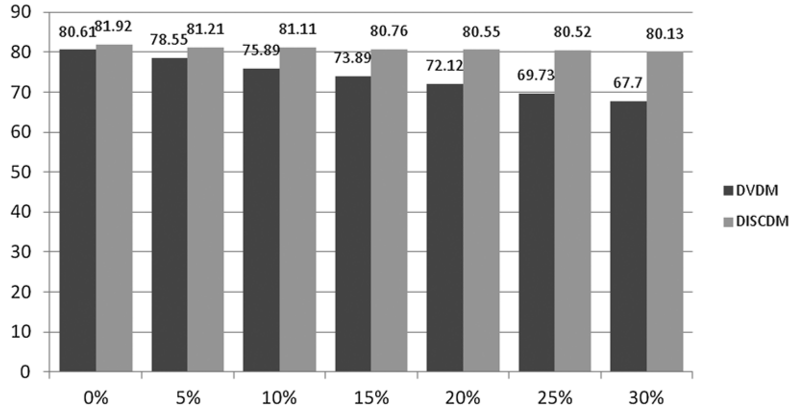


Fig. 1. The average classification accuracy of DVDM and DISCDM at different ratios of missing values.

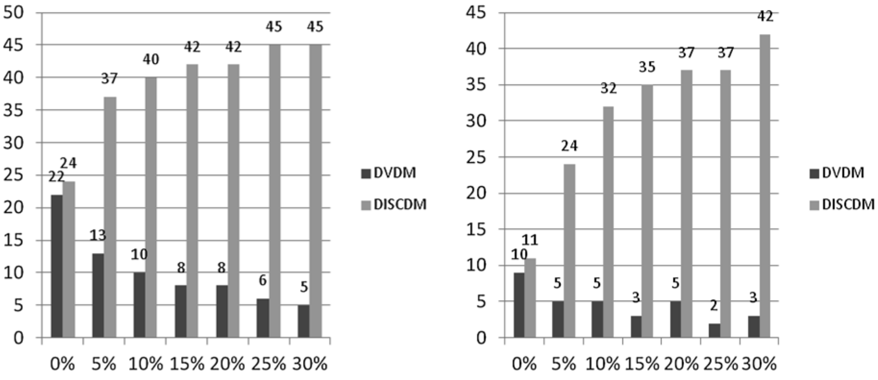


Fig. 2. Figure (A) shows the number of data sets on which DVDM and DISCDM achieved better results at different rates of missing values. Figure (B) shows the number of data sets on which DVDM and DISCDM achieved significantly better results at different rates of missing values.

The gap between the two measures also increases as the ratio of missing values increases with respect to the number of data sets on which they achieve better and significantly better results (see Fig. 2). The number of data sets on which DISCDM achieved better and significantly better accuracy almost always grew as the ratio of missing values increased. At 30% missing values DISCDM achieved better average accuracy on 45 data sets, 42 of which are significantly better with a 95% confidence level. On the other hand DVDM only achieved better results on 5 data sets, 3 of which was significantly better, at the 30% missing values level.

6. Tolerance to noise

In this section we compare DVDM and DISCDM with respect to their tolerance to noise, which is an important consideration given that real life data sets are rarely free of noise. We distinguish between two types of noise: noise in class attributes and noise in non-class attributes.

6.1. *Noise in non-class attributes*

We believe that noise in non-class attributes is more common than noise in the class attribute. This can be attributed to a general lack of care by record keepers in recording non-class attributes than when recording class attributes. Furthermore, there is only one class attribute and many non-class attributes in each data set, and the statistical chance of an error in a non-class attribute is therefore higher than the chances of an error in a class attribute.

To test the tolerance of DVDM and DISCDM to noise, we deliberately inserted noise in the data sets by replacing some randomly chosen attribute values with other randomly selected attribute values. The class attribute was excluded from this process. The test data

sets were not altered. We performed several sets of experiments using 10%, 20% and 30% noise rates. Due to the random nature of the process, and to ensure a statistically significant sample size, each experiment was repeated 5 times, performing 10-fold cross validation at each experiment. Table 4 summarizes the results.

The results show that DISCDM is less sensitive to non-class attribute noise than DVDM. The average classification accuracy of DVDM dropped from 80.61% at 0% noise to 72.88% at 30% noise, a drop of 7.73%. On the other hand DISCDM dropped from 81.92% at 0% noise to 76.62% at 30% noise, a drop of 5.29% (see Fig. 3). Moreover, the number of data sets on which DISCDM achieved higher and significantly higher accuracies, compared to DVDM increased consistently as the noise ratio increased from 0% to 30%

Table 4

The classification accuracy of DVDM and DISCDM in the presence of noise in non-class attributes

Data set	0%		10%		20%		30%	
	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM
Anneal	97.77	**99**	97.75	**98.66**	97.28	**98.38**	96.64	**97.8**
Anneal.ORIG	97.10	**97.66**	94.79	**96.86**	92.79	**95.03**	90.03	**92.56**
Arrhythmia	**74.34**	73.25	65.14	**73.73**	60.89	**65.9**	58.15	**60.59**
Audiology	71.84	**74.9**	59.01	**63.83**	49.66	**56.63**	43.25	**50.37**
Autos	58.57	**63.86**	56.05	**62.67**	56.11	**61.4**	57.17	**58.94**
Breast-cancer	67.87	**69.61**	68.98	**69.73**	67.16	**69.3**	**69.58**	67.61
Breast-w	**97.14**	96.71	**96.8**	96.77	96.37	**96.68**	96.23	**96.71**
Bridges ver1	53.45	**63**	45.42	**61.91**	38.55	**57.87**	33.65	**53.75**
Bridges ver2	48.18	**57.73**	38.96	**56.93**	36.13	**53.58**	30.02	**55.13**
Car	**90.39**	81.48	**81.42**	75.14	**75.85**	72.68	**72.86**	71.6
Colic	**85.05**	80.42	**83.25**	80.03	**79.76**	79.22	77.71	**78.15**
Colic.orig	65.2	**76.38**	62.39	**73.25**	59.12	**72.17**	**72.17**	**72.21**
Credit-a	80.87	**82.03**	**82.09**	82	80.26	**80.99**	79.01	**79.62**
Credit-g	72.3	**73.4**	71.7	**72.44**	69.88	**72.22**	68.98	**70.86**
Cylinder bands	59.07	**67.22**	60.26	**64.19**	57.11	**62.19**	56.33	**63.56**
Dermatology	**96.99**	96.72	94.53	**96.63**	89.17	**97.33**	86.06	**95.47**
Diabetes	**73.83**	73.69	**73.2**	72.4	**72.63**	72.57	70.44	**71.12**
Ecoli	**76.29**	75.98	74.5	**75.53**	**73.73**	73.15	67.02	**67.4**
Flags	**58.26**	57.61	48.79	**55.59**	45.39	**56.29**	42.71	**54.58**
Glass	**77.23**	72.51	**71.66**	71.3	63.88	**68.48**	60.7	**64.47**
Haberman	74.53	**74.85**	**73.68**	73.55	**72.24**	72.11	72.82	72.82
Heart-c	**85.45**	84.14	81.91	**82.25**	78.74	**82.64**	78.88	**81.13**
Heart-h	79.16	**83.57**	77.23	**78.52**	78.99	**80.15**	75.46	**78.52**
Heart-statlog	82.59	82.59	80.37	**82.44**	77.70	**81.33**	76.89	**79.85**
Hepatitis	84.38	**90.29**	85.03	**87.27**	83.68	**85.8**	82.61	**84.61**
Hypothyroid	**98.73**	98.49	96.93	**97.04**	95.93	**96.14**	94.97	**95.51**
Ionosphere	89.46	**90.88**	90.26	**91.51**	89.63	**91.00**	89.29	**90.89**
Labor	83	**88**	86.73	**91.87**	78.73	**92.27**	75.07	**88.2**
Letter	**93.79**	91.13	**90.40**	89.08	86.35	**86.56**	81.48	**83.20**
Liver-disorders	54.84	54.84	52.86	52.86	52.86	52.86	55.91	55.91
Lung-cancer	80	80	**74.67**	72.17	73.5	**74.67**	**74.17**	74
Lymph	81.81	**87.1**	78.54	**83.18**	77.11	**82.55**	73.87	**76.97**
Mushroom	99.93	99.93	99.91	**99.98**	99.7	**99.91**	99.15	**99.71**
Nursery	**88.12**	86.58	**76.41**	76.21	68.63	**69.28**	62.82	**64.81**
Optdigits	**97.38**	95.75	**96.47**	95.07	**95.44**	94.15	**94**	92.9
Pendigits	**98.35**	96.88	**97.56**	95.88	**96.17**	94.6	**94.49**	93.09
Primary tumor	38.89	**44.22**	36.73	**41.51**	32.22	**38.45**	30.79	**36.03**

Table 4
(Continued)

Data set	0%		10%		20%		30%	
	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM
Segment	**95.8**	94.81	94.04	**94.1**	92.44	**93.29**	90.72	**91.89**
Sick	**97.43**	97.16	95.94	**96.21**	95.42	**95.49**	94.72	**95.36**
Solar-flare_1	**97.52**	97.21	**97.53**	97.34	**97.28**	96.79	97.53	**97.59**
Solar-flare_2	**99.53**	99.06	**99.53**	99.17	**99.49**	99.38	**99.47**	99.45
Sonar	69.26	**80.74**	69.49	**77.3**	67.93	**72.9**	59.86	**67.44**
Soybean	89.14	**90.01**	82.27	**86.74**	71.64	**83.55**	63.57	**78.47**
Spambase	89.31	**91.24**	88.05	**90.47**	86.93	**89.89**	86.03	**89.11**
Splice	**93.57**	86.87	83.00	**85.49**	76.43	**82.36**	70.43	**79.97**
Trains	60	**70**	**74**	68	**66**	60	**64**	58
Vehicle	70.8	**72.22**	68.37	**69.69**	65.67	**68.29**	62.7	**66.24**
Vote	**95.39**	93.54	91.91	**93.13**	91.59	**92.41**	**91.07**	90.84
Vowel	62.22	**62.83**	53.78	**59.96**	51.19	**55.39**	45.05	**51.37**
Wine	**98.33**	97.75	95.89	**97.95**	93.99	**96.54**	90.62	**94.75**
Average	**80.61**	**81.92**	**77.92**	**80.11**	**75.11**	**78.46**	**72.88**	**76.62**
# Better	**22**	**24**	**15**	**35**	**10**	**38**	**8**	**40**
# Sig Better	**10**	**11**	**6**	**27**	**3**	**31**	**3**	**34**

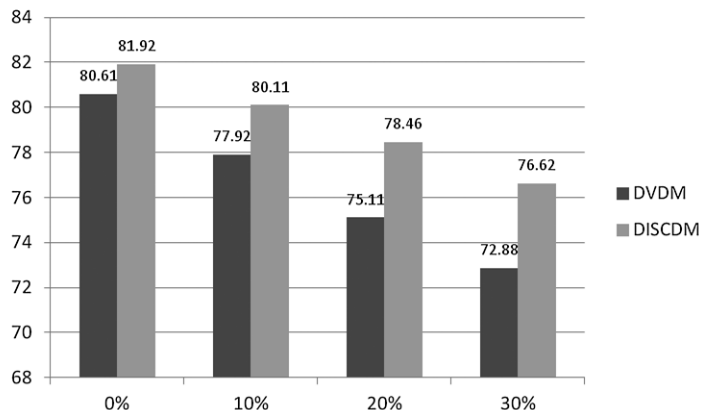


Fig. 3. The average classification accuracies of DVDM and DISCDM at different ratios of noise.

(see Fig. 4). The number of data sets on which DISCDM achieved significantly better results than DVDM, with a 95% confidence level, increased from 11 data sets at 0% noise to 34 data sets at 30% noise, while the number of data sets on which DVDM achieved significantly better results dropped from 10 at 0% noise to 3 data sets at 30% noise (see Fig. 4(B)).

Similarly, the number of data sets on which DISCDM achieved higher compared to DVDM rose from 24 data sets at 0% rate to 40 data sets at 30%, while the number of data sets on which DVDM achieved higher results than DISCDM dropped from 22 data sets at 0% noise to 8 data sets at 30% noise (see Fig. 4(A)).

It follows from the previous results that DISCDM is more tolerant to noise in the non-class attributes. This result is not surprising and can be explained by reference to the fact that the VDM formula is sensitive to noise in two places ($p(c|val_{\text{new}})$ and $p(c|val_{\text{train}})$), whereas the ISCDM formula is sensitive to noise in only one place ($p(val_{\text{new}}|class_{\text{train}})$).

Comparison of the results of this section with the results of the previous section reveals that noise has a greater negative effect than missing values on the classification accuracy of DISCDM. This is to be expected because noisy data is significantly more misleading than missing data. The average accuracy for DISCDM dropped from 81.92% at 0% missing values to 80.13%

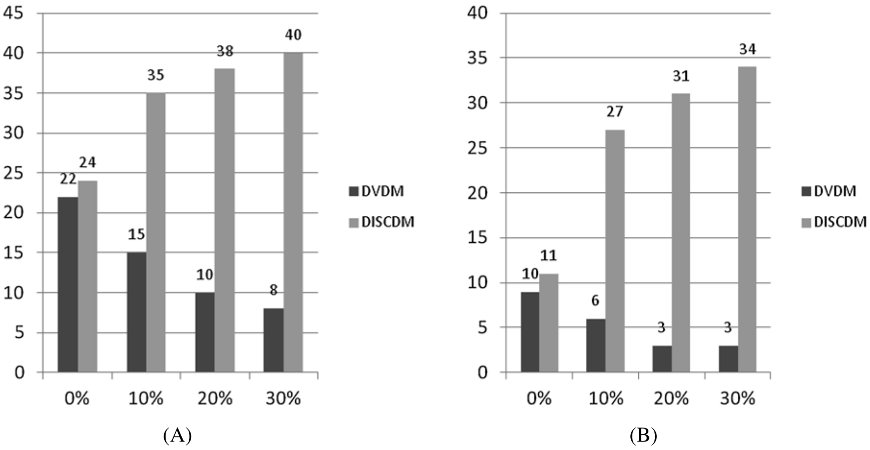


Fig. 4. Figure (A) shows the number of data sets on which DVDM and DISCDM achieved better accuracies at different rates of noise in non-class attributes. Figure (B) shows the number of data sets on which the two measures achieved significantly better results.

at 30% missing values, while it dropped from 81.92% at 0% noise to 76.62% at 30% noise. This suggests that eliminating potentially noisy attribute values in order that they can be handled as missing data may improve the classification accuracy of DISCDM.

On the other hand the results show that noisy data has less effect than missing attribute values on the classification accuracy of DVDM. Its classification accuracy dropped from 80.61% at 0% missing values and 0% noise to 67.70% at 30% missing values, and to 72.88% at 30% noise. This is somewhat surprising, since it would be reasonable to assume that noisy data would be more misleading and thus would have greater negative effect than missing values. The probable reason why missing values have greater negative effect on the results of the DVDM is that the way in which DVDM handles missing values, namely assuming that the distance is 1 if one of the values is missing, may be excessively cautious and harmful to accuracy.

6.2. *Noise in class attributes*

Although noise in the class attribute is less common than noise in non-class attributes, it is has a more serious effect on the classification accuracy. In this section, we study the effect of noise in the class attribute on the DVDM and DISCDM functions.

We deliberately inserted noise in training sets by replacing some randomly chosen values (classes) of the class attribute values with other random class values. We inserted noise only in the training set, leaving the test data set unchanged. We performed several sets of experiments using different rates of noise of 10%, 20% and 30%. Due to the random nature of the process and in order to ensure a statistically significant sample size, we repeated each experiment 5 times and performed 10-fold cross validation on each experiment. Table 5 summarizes the results.

It can be easily seen from Fig. 5 that the gap between DVDM and DISCDM in terms of the average classification accuracy decreased as the noise ratio increased. The average classification accuracy of DISCDM dropped from 81.92% at 0% noise to 71.36% at 30% noise, a drop of 10.56%. On the other hand, the average classification accuracy of DVDM dropped from 80.61% at 0% noise to 71.17% at 30% noise, a drop of 9.44%. This indicates that the accuracy of DISCDM decreased at a higher rate than the accuracy of DVDM as the noise ratio increased.

Figure 6 also shows that, although DISCDM achieved better results than DVDM at 0% noise, in terms of the number of data sets on which it achieved better and significantly better classification accuracy, the reverse was true at 30% noise, at which DVDM achieved better results to DISCDM on 29 data sets and achieved significantly better results on 20 of them while DISCDM achieved better results on 20 data sets, and achieved significantly better results on 14 of them. This also indicates that DISCDM is more sensitive than DVDM to noise in the class attribute.

The sensitivity of DISCDM to noise in the class attribute is due to the fact that DISCDM is based on $p(val_{\text{new}}|class_{\text{train}})$. This is computed using Eq. (6), in which noise in the class attribute affects both the nominator and the denominator, i.e. in two places. Conversely DVDM is based on Eq. (5) which is only vulnerable to this type of noise in the nominator.

Table 5
A summary of the classification accuracy obtained using DVDM and DISCDM at different noise ratios in the class attribute

Data set	0%		10%		20%		30%	
	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM
Anneal	97.77	**99**	**96.64**	96.46	**92.85**	92	**88.55**	86.68
Anneal.ORIG	97.10	**97.66**	**95.88**	94.70	**91.38**	88.89	**84.17**	82.67
Arrhythmia	**74.34**	73.25	**73.5**	71.33	**71.39**	68.37	**69.62**	65.85
Audiology	71.84	**74.9**	69.53	**69.61**	64.04	**65.02**	59.87	**59.9**
Autos	58.57	**63.86**	57.19	**61.49**	53.16	**56.8**	48.56	**57.81**
Breast-cancer	67.87	**69.61**	65.57	**66.64**	61.6	**66.67**	**62.74**	62.23
Breast-w	**97.14**	96.71	**95.51**	94.43	91.96	**92.05**	89.22	**90.71**
Bridges_ver1	53.45	**63**	52.56	**57.02**	48.2	**56.8**	42.25	**52.44**
Bridges_ver2	48.18	**57.73**	46.44	**55.2**	44.16	**52.38**	41.31	**50.51**
Car	**90.39**	81.48	**88.94**	78.11	**84.32**	75.57	**82**	73.84
Colic	**85.05**	80.42	**82.03**	78.94	**77.65**	74.72	**73.91**	73.13
Colic.orig	65.2	**76.38**	60.86	**72.5**	56.89	**69.83**	54.61	**68.52**
Credit-a	80.87	**82.03**	77.97	**78.61**	**76.03**	75.94	**71.91**	71.59
Credit-g	72.3	**73.4**	68.66	**70.96**	67.12	**69.04**	64.12	**65.76**
Cylinder-bands	59.07	**67.22**	55.26	**65.85**	55.93	**64.52**	55.15	**63.07**
Dermatology	**96.99**	96.72	**95.46**	94.06	89.11	**89.91**	81.05	**85.16**
Diabetes	**73.83**	73.69	**72.42**	70.49	**69.87**	66.09	**68.91**	65.18
Ecoli	**76.29**	75.98	**77.35**	76.84	**78.01**	75.64	**76.92**	74.36
Flags	**58.26**	57.61	52.81	**53.54**	**49.69**	49.09	46	**47.12**
Glass	**77.23**	72.51	**75.91**	71.39	**74.79**	67.54	**71.02**	64.25
Haberman	74.53	**74.85**	73.42	**73.81**	73.03	**73.22**	70.3	**70.56**
Heart-c	**85.45**	84.14	**82.48**	80.64	78.72	**78.94**	**75.7**	73.18
Heart-h	79.16	**83.57**	77.97	**80.36**	**78.86**	78.03	**78.3**	75.36
Heart-statlog	82.59	82.59	**79.04**	73.78	**74.22**	64.89	**69.33**	64.52
Hepatitis	84.38	**90.29**	81	**84.38**	77.62	**80.77**	**76.18**	76.17
Hypothyroid	**98.73**	98.49	**98.28**	97.03	**96.85**	94.96	**94.64**	91.65
Ionosphere	89.46	**90.88**	86.72	**89.80**	80.74	**86.96**	71.90	**78.61**
Labor	83	**88**	81.73	**89.67**	78.67	**87.13**	71.4	**84.27**
Letter	**93.79**	91.13	**91.61**	89.28	**86.92**	85.32	**79.99**	79.47
Liver-disorders	54.84	54.84	55.51	55.51	54.55	54.55	61.46	61.46
Lung-cancer	80	80	70	**71.33**	67.83	67.83	**68.83**	65.5
Lymph	81.81	**87.1**	78.15	**81.74**	75.11	**78.49**	69.42	**69.86**
Mushroom	99.93	99.93	**97.57**	97.16	**93.02**	92.24	**86.6**	86.35
Nursery	**88.12**	86.58	**86.75**	82.02	**82.82**	78.27	**78.14**	74.1
Optdigits	**97.38**	95.75	**95.15**	94.19	91.01	**92.07**	84.86	**89.13**
Pendigits	**98.35**	96.88	**96.56**	95.64	92.07	**92.96**	86.67	**89.61**
Primary-tumor	38.89	**44.22**	39.96	**41.26**	38.2	38.2	36.84	**37.36**
Segment	**95.8**	94.81	**93.66**	93.21	89.34	**89.72**	84.82	**85.9**
Sick	**97.43**	97.16	**96.67**	94.94	**95.21**	92.61	**93.15**	90.21
Solar-flare_1	**97.52**	97.21	**94.79**	93.63	**90.59**	87.81	**82.21**	79.48
Solar-flare_2	**99.53**	99.06	**98.68**	97.56	**97.35**	95.14	**93.70**	91.78
Sonar	69.26	**80.74**	67.84	**71.63**	**69.03**	65.56	**65.94**	59.29
Soybean	89.14	**90.01**	**87.18**	85.21	**81.23**	77.91	**76.41**	75.2
Spambase	89.31	**91.24**	**86.41**	85.73	**80.77**	78.96	**72.28**	70.88
Splice	**93.57**	86.87	**89.99**	83.70	**80.16**	77.98	52.67	**70.58**
Trains	60	**70**	60	**70**	62	**74**	**56**	52

Table 5

(Continued)

Data set	0%		10%		20%		30%	
	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM	DVDM	DISCDM
Vehicle	70.8	**72.22**	**69.33**	68.95	65.93	**67.47**	62.19	**62.81**
Vote	**95.39**	93.54	**93.36**	80	**89.87**	71.66	**84.72**	65.25
Vowel	62.22	**62.83**	59.35	**61.66**	**58.14**	57.72	**55.56**	55.21
Wine	**98.33**	97.75	**96.11**	93.58	**91.24**	85.12	**86.4**	81.48
Average	80.61	81.92	78.52	78.71	75.39	75.27	71.17	71.36
# Better	22	24	29	21	26	21	29	20
# Sig Better	10	11	24	13	21	15	20	14

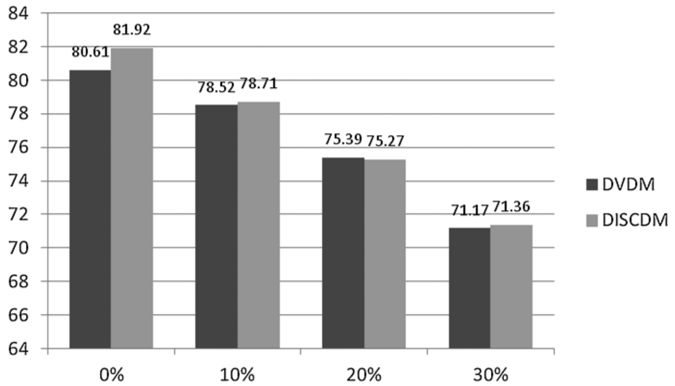
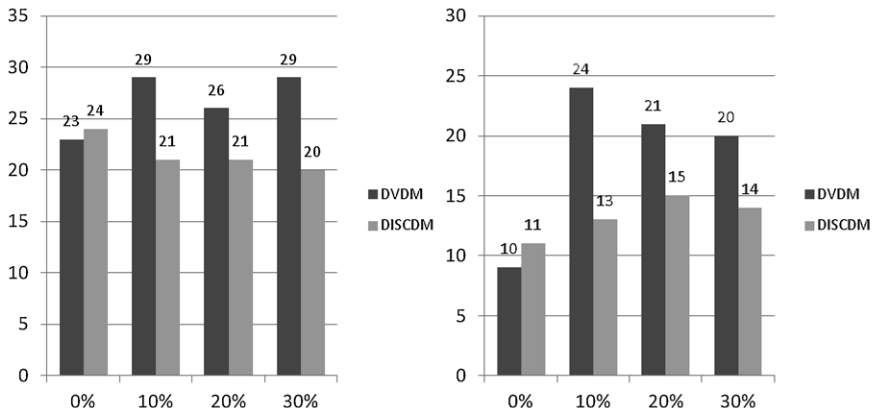


Fig. 5. The average classification accuracy of DVDM and DISCDM at different ratios of noise in the class attribute.



(A) (B)

Fig. 6. Figure (A) shows the number of data sets on which DVDM and DISCDM achieved better accuracies at different ratios of noise in the class attribute. Figure (B) shows the number of data sets on which the two measures achieved significantly better results.

6.2.1. Value Specific Distance Measure (VSDM)

The previous discussion leads to the conclusion that we may get a distance measure that is more tolerant of noise in a class attribute than either DISCDM or DVDM if we replace the term $p(val_{\text{new}}|class_{\text{train}})$ in the DISCDM definition with the term $p(class_{\text{train}}|val_{\text{new}})$. We call the resulting new measure Value Specific Distance Measure (VSDM), which is defined as follows:

$$\mathrm{VSDM}(val_{\text{new}}, val_{\text{train}}) = 1 - p(class_{\text{train}}|val_{\text{new}}). \quad (10)$$

The measure defines the distance between two nominal values, val_{new} and val_{train}, as the probability of a class, other than $class_{\text{train}}$, occurring, given val_{new}. This measure is expected to be more tolerant of noise in the class attribute than DISCDM because, like DVDM, noise in a class attribute would only affect the value of the nominator in the formula for computing $p(class_{\text{train}}|val_{\text{new}})$. It is also expected to be more tolerant of noise than VDM, because VDM uses not only the term $p(class_c|val_{\text{new}})$, but also the term $p(class_c|val_{\text{train}})$.

To test the tolerance of VSDM to noise in the class attribute against the tolerance of DVDM, similar experiments to those described in the previous section were performed. Table 6 summarizes the results. Since we discretized all ordinal attributes at a preprocessing stage, we prefixed the names of the measures with "D". At 0% noise ratio DVDM and DVSDM

Table 6
A summary of the classification accuracy obtained using DVDM and DVSDM at different noise ratios in the class attribute

Data set	0%		10%		20%		30%	
	DVDM	DVSDM	DVDM	DVSDM	DVDM	DVSDM	DVDM	DVSDM
Anneal	**97.77**	94.99	**96.7**	94.62	92.81	**93.61**	88.29	**92.25**
Anneal.ORIG	**97.10**	95.21	**95.26**	94.06	91.45	**92.56**	83.62	**89.49**
Audiology	**71.84**	69.15	**69.25**	67.29	65.6	**65.8**	60.39	**61.38**
Arrhythmia	**74.34**	63.52	**73.46**	63.6	**70.82**	64	**68.94**	63.64
Autos	58.57	**63.26**	55.03	**60.77**	50.79	**58.9**	49.85	**56.63**
Breast-cancer	67.87	**69.95**	64.63	**69.25**	61.3	**68.96**	62.7	**68.83**
Breast-w	**97.14**	96.43	**95.28**	94.39	**92.97**	91.48	90.25	**90.36**
Bridges_ver1	53.45	**55.27**	50.84	**55.67**	48.8	**52.49**	45.8	**49.56**
Bridges_ver2	48.18	**53.55**	48.93	**52.76**	40.36	**51.13**	42.95	**51.29**
Car	**90.39**	70.01	**88.57**	70.01	**84.86**	70.01	**81.45**	70.01
Colic	**85.05**	79.08	**81.55**	75.04	**77.85**	75.7	73.64	**74.23**
Colic.orig	65.2	**71.16**	60.54	**70.66**	56	**69.15**	53.46	**69.7**
Credit-a	**80.87**	73.19	**79.36**	66.93	**76.43**	63.3	**72.35**	61.25
Credit-g	**72.3**	70	68.84	**70.3**	66.86	**70.46**	63.9	**70.42**
Cylinder-bands	59.07	**65.37**	55.37	**63.74**	52.56	**62.52**	53.96	**61.44**
Dermatology	**96.99**	96.72	**94.64**	93.63	**88.25**	88.15	82.00	**83.55**
Diabetes	**73.83**	69.13	**72.37**	70.13	**70.07**	68.61	**68.77**	68.46
Ecoli	**76.29**	75.97	**78.74**	77.71	**75.31**	74.06	**76.58**	75.27
Flags	**58.26**	56.08	54.23	**55.79**	48.69	**55.99**	46.32	**54.02**
Glass	**77.23**	76.73	**76.3**	76.08	**72.74**	71.12	**71.42**	69.52
Haberman	74.53	**74.85**	72.7	**72.96**	**71.92**	71.59	70.48	**70.68**
Heart-c	**85.45**	83.17	**82.02**	81.06	**79.45**	79.06	77.8	**77.99**
Heart-h	79.16	**79.75**	**78.93**	78.65	**78.26**	77.1	**77.29**	75.3
Heart-statlog	82.59	**83.33**	**79.41**	73.70	**73.93**	66.37	**69.33**	60.37
Hepatitis	**84.38**	80.96	**82.33**	80.43	77.76	**81.66**	72.02	**81.96**
Hypothyroid	**98.73**	94.25	**98.48**	94.19	**96.82**	93.91	**94.60**	93.23
Ionosphere	89.46	**92.59**	87.30	**90.94**	80.14	**87.18**	70.55	**84.21**
Labor	83	**94.67**	78.8	**86.2**	74.73	**83.33**	68.27	**80.8**
Letter	**93.79**	90.76	**91.52**	88.78	**86.96**	84.80	**80.05**	79.25
Liver-disorders	54.84	54.84	55.21	55.21	55.14	55.14	56.18	56.18
Lung-cancer	**80**	70	**76.17**	72	70.17	**70.83**	62.67	**72.17**
Lymph	**81.81**	81.76	79.9	79.9	74.96	**78.02**	68.38	**77.33**
Mushroom	**99.93**	99.88	**97.65**	95.84	92.74	**93.46**	87.14	**91.2**
Nursery	**88.12**	79.99	**86.72**	79.17	**82.52**	76.9	**77.65**	75.57
Optdigits	**97.38**	94.89	**95.2**	93.29	90.95	**91.1**	85.43	**87.73**

Table 6
(Continued)

Data set	0%		10%		20%		30%	
	DVDM	DVSDM	DVDM	DVSDM	DVDM	DVSDM	DVDM	DVSDM
Pendigits	**98.35**	96.82	**96.6**	95.74	92.19	**93.56**	86.52	**90.35**
Primary-tumor	38.89	**38.89**	39.21	**41.38**	37.42	**39.96**	36.79	**39.5**
Segment	**95.8**	95.15	93.71	**93.76**	89.77	**91.49**	84.62	**88.42**
Sick	**97.43**	95.84	**96.7**	95.5	**95.19**	94.94	93.21	**93.8**
Solar-flare_1	97.52	**97.83**	94.67	**97.72**	89.59	**96.66**	82.32	**93.87**
Solar-flare_2	99.53	99.53	98.91	**99.49**	96.94	**98.95**	93.24	**97.35**
Sonar	69.26	**71.24**	**67.84**	66.84	**68.89**	61.18	**67.02**	51.62
Soybean	**89.14**	82.85	**85.53**	80.39	**82.59**	76.67	**77.16**	73.53
Spambase	**89.31**	85.44	**86.66**	80.02	**81.01**	77.92	73.09	**73.88**
Splice	**93.57**	54.98	**89.84**	55.22	**80.43**	54.84	**56.71**	54.73
Trains	60	**70**	60	**70**	**62**	56	62	62
Vehicle	70.8	**72.46**	69.21	**70.41**	65.86	**66.64**	62.53	**63.55**
Vote	**95.39**	91.69	**93.22**	83.99	**90.55**	81.79	**84.64**	76.55
Vowel	**62.22**	60.91	**59.86**	59.21	**56.67**	55.92	**53.19**	52.4
Wine	**98.33**	98.3	**96.11**	94.18	**93.09**	91.19	**86.56**	84.33
Average	**80.61**	**78.70**	**78.61**	**76.97**	**75.06**	**74.72**	**71.08**	**72.82**
# Better	**32**	**16**	**32**	**17**	**26**	**25**	**18**	**33**
# Sig. better	**18**	**3**	**20**	**13**	**15**	**16**	**12**	**22**

achieved 80.061% and 78.70% average accuracy respectively. The difference is 1.91% in favor of DVDM. However, as noise ratio increases DVSDM gradually becomes more accurate than DVDM. At 30% noise, the average accuracy of DVSDM becomes higher than the average accuracy of DVDM, where DVDM and DVSDM achieved 71.08% and 72.82% average accuracies respectively. The difference is 1.74% in favor of DVSDM. This indicates that DVSDM is clearly less sensitive to noise in the class attribute than DVDM.

This conclusion is further supported by the fact that the number of data sets on which DVSDM achieved better and significantly better results (last two rows in Table 6) than DVDM increased as the noise ratio was increased. At 0% noise DVDM achieved better results than DVSDM on 32 data sets, of which 18 data sets were significantly better, while DVSDM achieved better results on 16 data sets, of which only 3 were significantly better. At 30% noise the reverse is true; DVDM achieved better results than DVSDM on only 18 data sets, of which 12 are significantly better, while DVSDM achieved better results on 33 data sets, of which 22 are significantly better. An alternative solution to the problem of the excessive sensitivity of DISCDM to noisy class attributes may be found by eliminating the noisy instances at a preprocessing stage, using a noise filtering algorithm such as the ENN, RENN [22] and All-kNN [23] algorithms. Filtering out noisy instances using these algorithms was shown to improve the training speed of neural networks and their generalization accuracy [8,9].

7. Conclusion

This work proposed several distance measures for nominal attribute values that make use of the class of a training example against which a new instance is being compared. As with the VDM, the proposed measures only work with labeled training data which makes them unsuitable for unsupervised learning methods. Together with VDM, the proposed measures provide a pool of diverse distance functions that can be used for different application domains that may require different inductive biases. Furthermore, one of the proposed measures, namely ISCDM and its discretized version DISCDM, gave better results than VDM and its discretized version DVDM, in terms of the average classification accuracy on 50 data sets and in terms of the overall number of data sets on which it achieved better accuracy. DISCDM, also, proved to be much less sensitive to noise in non-class attributes and to missing data in a training set than DVDM. Although DISCDM is less sensitive to noise in non-class attributes than

DVDM, it has been found to be more sensitive to noise than to missing data in non-class attributes. Therefore, it might be beneficial to devise a method for identifying and eliminating noisy data in non-class attributes and dealing with them as missing values. DISCDM was, however, found to be more sensitive to noise in class attributes than DVDM; therefore DVSDM has been proposed and, as expected, proved to be less sensitive to noise in the values of a class attribute. Another possible solution to the noise-sensitivity problem is to filter out noisy instances using a noise filtering algorithm at a preprocessing stage. As mentioned before, the distance function used influences the inductive bias of the learning algorithm; therefore there is no single distance function that is consistently the most accurate across all application domains. This stresses the need for a selection method that could be used to determine the most appropriate distance function for a given application domain. Current research is exploring such selection methods.

Acknowledgements

This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University. The author is grateful for this support.

References

[1] D.W. Aha, D. Kibler and M.K. Albert, Instance-based learning algorithms, *Machine Learning* **6**(1) (1991), 37–66.

[2] A. Argentini and E. Blanzieri, Neighborhood counting measure metric and minimum risk metric: An empirical comparison, Information Engineering and Computer Science, Technical Report DISI-08-057, Univ. of Toronto.

[3] C. Blake and C. Merz, UCI repository of machine learning databases [online], Department of Information and Computer Science, University of California, Irvine, CA, 1998, available at: http://www.ics.uci.edu/˜mlearn/MLRepository.html.

[4] E. Blanzieri and F. Ricci, Probability based metrics for nearest neighbor classification and case-based reasoning, *Lecture Notes in Computer Science* **1650** (1999), 14–29.

[5] M. Boullé, Modl: A Bayes optimal discretization method for continuous attributes, *Machine Learning* **65**(1) (2006), 131–165.

[6] M.D. Buhmann and M.D. Buhmann, *Radial Basis Functions*, Cambridge Univ. Press, New York, NY, USA, 2003.

[7] J. Dougherty, R. Kohavi and M. Sahami, Supervised and unsupervised discretization of continuous features, in: *Proceedings of the Twelfth International Conference on Machine Learning*, Morgan Kaufmann Publishers, 1995, pp. 194–202.

[8] K. El Hindi and M. Al-Akhras, Eliminating border instance to avoid overfitting, in: *International Conference Intelligent Systems and Agents*, Algarve, Portugal, 2009.

[9] K. El Hindi and M. Al-Akhras, Smoothing decision boundaries to avoid overfitting in neural network training, *Neural Network World* **21** (2011), 311–325.

[10] U. Fayyad and K. Irani, Multi-interval discretization of continuous values attributes for classification learning, in: *Proceedings of 13th International Joint Conference on Artificial Intelligence*, 1993.

[11] A. Globerson and S. Roweis, Metric learning by collapsing classes, *Advances in Neural Information Processing Systems* **18** (2006), 451.

[12] T. Kohonen, M.R. Schroeder and T.S. Huang, eds, *Self-Organizing Maps*, 3rd edn, Springer-Verlag, New York, Secaucus, NJ, USA, 2001.

[13] S.P. Lloyd, Least square quantization in pcm, *IEEE Transaction on Information Theory* **28**(2) (1982), 129–137.

[14] T.M. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, USA, 1997.

[15] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[16] C. Schaffer, A conservation law for generalization performance, in: *The Eleventh International Conference on Machine Learning*, 1994.

[17] C. Stanfill and D. Waltz, Toward memory-based reasoning, *Communication of ACM* **9** (1986), 1213–1228.

[18] H. Wang, Nearest neighbors by neighborhood counting, *IEEE Trans. Pattern Analysis and Machine Intelligence* **28** (2006), 942–953.

[19] H. Wang, Neighborhood counting measure and minimum risk metric, *IEEE Trans. Pattern Analysis and Machine Intelligence* **32** (2010), 766–768.

[20] H. Wang and W. Dubitzky, A flexible and robust similarity measure based on contextual probability, in: *International Joint Conference on Artificial Intelligence*, Vol. 19, Lawrence Erlbaum Associates, 2005, p. 27.

[21] K.Q. Weinberge and L.K. Saul, Distance metric learning for large margin, *Journal of Machine Learning Research* **10** (2009), 207–244.

[22] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man, and Cybernetics* **2**(3) (1972), 408–421.

[23] D.L. Wilson, An experiment with the edited nearest-neighbor rule, *IEEE Transactions on Systems, Man, and Cybernetics* **6**(6) (1976), 448–452.

[24] D.R. Wilson and T.R. Martinez, Improved heterogeneous distance functions, 1997, arXiv preprint cs/9701101.

[25] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005.