ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

K-Nearest Neighbours based on Mutual Information for Incomplete Data Classification

Pedro J. García-Laencina[1], José-Luis Sancho-Gómez[1],
Anibal R. Figueiras-Vidal[2] and Michel Verleysen[3] *

1- Universidad Politécnica de Cartagena - Dpto. TIC
Plaza del Hospital, 1, 30202, Cartagena (Murcia) - Spain.

2- Universidad Carlos III de Madrid - Dpto. TSC
Avda. de la Universidad, 30, 28911, Leganés (Madrid) - Spain.

3- Université catholique de Louvain - Machine Learning Group, DICE
3 place du Levant, 1348, Louvain-la-Neuve - Belgium.

Abstract. Incomplete data is a common drawback that machine learning techniques need to deal with when solving real-life classification tasks. One of the most popular procedures for solving this kind of problems is the K-nearest neighbours (KNN) algorithm. In this paper, we present a weighted KNN approach using mutual information to impute and classify incomplete input data. Numerical results on both artificial and real data are given to demonstrate the effectiveness of the proposed method.

1 Introduction

Missing values in data sets may have different origins such as death of patients, equipment malfunctions, refusal of respondents to answer certain questions, and so on [1]. The correct treatment of missing values is an important step for solving pattern recognition tasks. This paper is focused on classification problems given an input data set with missing values. We present a variant of the K-nearest neighbours (KNN) algorithm based on Mutual Information (MI). This approach finds the nearest neighbours using a distance measure that considers the relevance of each input feature with the classification task. The aim of this paper is to analyze the performance of the KNN algorithm on incomplete data classification tasks, and to improve its effectiveness using the MI.

The remaining of this work is organized as follows. Section 2 shows the notation used in this paper. Missing data imputation with the standard KNN approach and the proposed KNN algorithm based on MI concept are both described in Section 3. Section 4 proposes a KNN procedure using MI for performing decision tasks. Next, in Section 5, experimental results on both real and artificial data sets are shown. Finally, Section 6 presents the main conclusions and future related works.

*This work is partially supported by Ministerio de Educación y Ciencia under grant TEC2006-13338/TCM, and by Fundación Séneca (Consejería de Educación y Cultura de Murcia) under grant 03122/PI/05.

2 Classifying with Missing Data

Let us assume a set of N labeled incomplete patterns[1], $\mathcal{D} = \{(\mathbf{x}^i, t^i, \mathbf{m}^i)\}_{i=1}^{N}$, where $\mathbf{x}^i$ is the i-th input vector with n features ($\mathbf{x}^i = \{x_j^i\}_{j=1}^{n}$), labeled as t^i, with c possible classes; $\mathbf{m}^i$ are binary variables taking value 1 if x^i is unknown, 0 otherwise. This work assumes that the missing data is missing completely at random (MCAR), meaning that the probability that an input feature x_j^i is missing is unrelated to the value of x_j^i or to the value of any other variables [1].

In general, pattern classification with missing data concerns two different problems, *handling missing values* and *pattern classification.* This work analyzes the performance of the KNN algorithm to impute and classify incomplete input data. Using this method, in a first stage, the missing values are imputed with KNN, and after that, the classification accuracy is performed by a KNN classifier using the edited set (complete patterns and imputed cases).

3 Imputation with the KNN algorithm

3.1 KNNimpute algorithm

The algorithm **KNNimpute** selects the K_I closest cases from the training cases with known values in the attributes to be imputed, such that they minimise some distance measure [2,3]. We use K_I to refer to the K nearest neighbours used for imputation. Once the K_I nearest neighbours have been found, a replacement value to substitute for the missing attribute value must be estimated. How the replacement value is calculated depends on the type of data; the mode can be used for categorical data and the mean for continuous data. It has been shown that this method provides a robust procedure for missing data estimation [2,3]. Its major drawback is that whenever the KNNimpute looks for the most similar instances, the algorithm has to search through all the data set. This limitation can be critical for large databases. To apply the KNN approach to unknown data, the following two issues should be addressed.

3.1.1 Measuring Distance

The distance between two incomplete cases $\mathbf{x}^a$ and $\mathbf{x}^b$ is denoted as $d(\mathbf{x}^a, \mathbf{x}^b)$. This work uses an heterogeneous distance function that computes different attribute distance measures on different types of attributes. In particular, the *Heterogeneous Euclidean-Overlap Metric* (HEOM) is used [3]:

$$d_H(\mathbf{x}^a, \mathbf{x}^b) = \sqrt{\sum_{j=1}^{n} d_j(x_j^a, x_j^b)^2}, \qquad (1)$$

[1] In the following, the terms pattern, case, input vector and instance are used as synonyms.

where $d_j(x_j^a, x_j^b)$ is the distance between $\mathbf{x}^a$ and $\mathbf{x}^b$ on its j-th attribute:

$$d_j(x_j^a, x_j^b) = \begin{cases} 1, & (1 - m_j^a)(1 - m_j^b) = 0 \\ d_O(x_j^a, x_j^b), & x_j \text{ is a categorical attribute} \\ d_N(x_j^a, x_j^b), & x_j \text{ is a numerical attribute.} \end{cases} \tag{2}$$

Unknown data are handled by returning a distance value of 1 (i.e., maximal distance) if either of the input values is unknown. The *overlap* distance function d_O assigns a value of 0 if the categorical attributes are the same, otherwise a distance value of 1. The *range normalized difference* distance function d_N is

$$d_N(x_j^a, x_j^b) = \frac{|x_j^a - x_j^b|}{max(x_j) - min(x_j)}, \tag{3}$$

where $max(x_j)$ and $min(x_j)$ are respectively the maximum and minimum values observed in the training set for the numerical attribute x_j.

3.1.2 Selection of the Neighbourhood

The K_I closest cases can be selected on the instances without any missing value, or by considering only instances with non-missing entries in the incomplete attribute to be imputed. The second option is more recommended [2]. Optimal K_I can be chosen by the cross-validation method.

3.2 Weighted Distance using Mutual Information

Selecting features before building a predictive model is an useful approach. One possible way to select the features that are relevant for a classification or function approximation problem, is to measure the Mutual Information (MI) between each feature individually and the objective task [4, 5]. Consider that α_j represents the MI measured between the j-th input attribute and the target class t. MI can be interpreted as the amount of information that x_j contains about t. The higher is the value of α_j, the more relevant is x_j for the classification task [5]. Following the feature selection concept, the **MI-KNNimpute** approach is proposed using a *MI-weighted distance metric.* The MI-weighted distance between two incomplete cases $\mathbf{x}^a$ and $\mathbf{x}^b$ is computed by

$$d_I(\mathbf{x}^a, \mathbf{x}^b) = \sqrt{\sum_{j=1}^{n} \frac{\alpha_j d_j(x_j^a, x_j^b)}{\sum_{j'=1}^{n} \alpha_{j'}}^2}, \tag{4}$$

where d_j is the heterogeneous distance defined by (2). The MI estimate used in this paper results from a Parzen window approach [4], by considering only training cases with known values in the attribute of interest. In contrast to the standard KNNimpute, this method selects the K_I nearest cases considering the input attribute relevance to the target class. By doing this, it provides a missing data estimation oriented to solve the classification task.

4 Exploiting MI to Classify with KNN algorithm

According to the KNN algorithm, an unclassified pattern is assigned to the class represented by a majority of its K_C nearest neighbours from the training set (**KNNclassify**). Some approaches have been proposed based on assigning different weights to the K_C neighbours based on their distances to $\mathbf{x}$, with closer neighbours having greater weights [6]. Following this idea, we also propose an enhanced version of the KNNclassify method using the *MI-weighted distance metric*. It is called **MI-KNNclassify**. Let $d_I(\cdot)$ be the distance measure based on the MI concept defined by (4), and $[\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^{K_C}]$ be the K_C nearest neighbours of $\mathbf{x}$ arranged in increasing order of $d(\mathbf{v}^k, \mathbf{x})$, with $k = 1, 2, \ldots, K_C$. So $\mathbf{v^1}$ is the closest neighbour of $\mathbf{x}$. MI-KNNclassify assigns to the k-th nearest neighbour $\mathbf{v}^k$ a weight $\beta_k(\mathbf{x})$ given by,

$$\beta_k(\mathbf{x}) = \begin{cases} \frac{d_I(\mathbf{v}^{K_C},\mathbf{x}) - d_I(\mathbf{v}^k,\mathbf{x})}{d_I(\mathbf{v}^{K_C},\mathbf{x}) - d_I(\mathbf{v}^1,\mathbf{x})}, & \text{if } d_I(\mathbf{v}^{K_C},\mathbf{x}) \neq d_I(\mathbf{v}^1,\mathbf{x}) \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

Pattern $\mathbf{x}$ is assigned to the class for which the weights of the representatives among the K_C nearest neighbours sum to the largest value. Doing this, β_k is computed considering how relevant are the input features for classification.

5 Experimental Results

In order to compare the proposed method with the standard KNN approach, we have chosen a complete problem[2], *Telugu*, and an incomplete data set[3], *Hepatitis*. The main reason for using a complete problem is to measure the influence of the percentage of artificially inserted missing data on the classification accuracy. The tested methods are both trained using the same *training*, *validation*, and *test* sets obtained by the stratified 10-fold cross validation method. For each fold in the *Telugu* set, a percentage of missing data (5%, 10%, 20%, 30% and 40%) are inserted into the selected attributes in a completely at random manner [1]. In all tested problems, firstly, the test set is classified by KNNclassify and MI-KNNclassify without estimating the missing values. Next, we check whether the classification performance can be improved imputing the missing values by KNNimpute and MI-KNNimpute. The optimal values of K to impute (K_I) and to classify (K_C) are selected by cross-validation.

5.1 Telugu Data

Telugu is an Indian vowel recognition problem. This data set is a six-class problem composed of 871 cases with three real attributes. Before missing data are inserted, the MI between each input feature and the target class are evaluated: $\alpha_1 = 1.043$, $\alpha_2 = 1.381$, and $\alpha_3 = 0.829$. It is critical to observe that not all the

[2]http://www.isical.ac.in/~sushmita/patterns/
[3]http://mlearn.ics.uci.edu/MLRepository.html

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

features equally contribute to solve the problem. Considering this fact, missing values have been randomly introduced in the attributes x_1 and x_2. Table 1 shows the classification results obtained without imputation. We test the following values for K_I and K_C: 1, 5, 10, 15, 20, 30, 40 and 50. For each percentage of missing data, K_C has been selected by cross-validation. We can see how MI-KNNclassify clearly outperforms the standard KNN method in all simulations. After that, we test wheter a missing data imputation provides better classification results. The different values of K_I and K_C have also been selected by cross-validation (using the classification accuracy in the validation set). As we can observe in Table 2, a missing data estimation helps to improve the classifier accuracy in all simulations. With low percentage of missing data (from 5% to 30%), the combined approach MI-KNNclassify and MI-KNNimpute outperforms the other tested procedures. The advantage of using MI-KNNimpute is not so clear for a higher percentage of missing data, as we can see with 40%. In this case, KNNimpute is a better approach for imputation. It is due to the fact that the estimated MI is not so accurate (we have less information to compute the values of MI, i.e., α_j).

	Missing data (%) in x_1 and x_2				
	5%	10%	20%	30%	40%
KNNclassify	83.69	82.32	76.11	72.77	68.98
MI-KNNclassify	**84.61**	**83.23**	**78.30**	**75.53**	**70.00**

Table 1: Test classification accuracy (%) for *Telugu* data using KNNclassify and MI-KNNclassify (without imputation), for different percentages of missing values in the attributes x_1 and x_2. The best results are shown in bold face.

Missing Data	**KNNclassify**		**MI-KNNclassify**	
	KNNimpute	**MI-KNNimpute**	**KNNimpute**	**MI-KNNimpute**
5%	85.29	86.09	85.53	86.21
10%	83.78	84.25	84.38	84.60
20%	78.85	79.02	78.76	79.23
30%	75.18	75.51	75.64	75.73
40%	69.91	69.60	69.99	69.69

Table 2: Test classification accuracy (%) for *Telugu* data using KNNclassify and MI-KNNclassify after imputation in the attributes x_1 and x_2 is done.

5.2 Hepatitis Data

This data set is composed of 155 patients described by 19 attributes. Among these patients, 32 patients died of hepatitis while the remaining ones survived. There are 80 incomplete cases and 5.7% of the input data are missing.

Table 3 shows the test classification accuracy (%) without missing data imputation obtained by KNNclassify and MI-KNNclassify for different K_C values. We test the following values for K_I and K_C: 1, 5, 10, 15, and 20. As

	Number of nearest neighbours (K_C)				
	1	5	10	15	20
KNNclassify	80.09	81.39	**80.02**	81.28	81.28
MI-KNNclassify	76.06	79.24	83.12	**83.20**	81.99

Table 3: Test classification accuracy (%) for *Hepatitis* data without imputation vs. Number of nearest neighbours (K_C) in KNNclassify and MI-KNNclassify. Selected values by cross-validation are shown in bold face.

it can be observed in Table 3, the MI-KNN approach outperforms the KNNclassify method. After that, unknown values are estimated using KNNimpute and MI-KNNimpute. The optimal values of K_I and K_C have been selected by cross-validation. The test classification results after imputing missing values are the following: **84.56** (KNNclassify + KNNimpute), **85.08** (KNNclassify + MI-KNNimpute), **85.70** (MI-KNNclassify + KNNimpute) and **86.25** (MI-KNNclassify + MI-KNNimpute). It is clear to see that a missing data imputation using the MI concept helps to improve the classification accuracy.

6 Conclusions

In this work, we have established a new KNN method to classify and impute incomplete data based on the MI concept. The proposed procedure selects the K nearest cases considering the input attribute relevance to the target class. It is done using a MI-weighted distance metric. During the missing data estimation stage, this method provides an imputation oriented to solve the classification task. Moreover, an effective KNN classifier based on the MI-weighted distance has also been proposed. Experimental results on both artificial and real incomplete databases show the usefulness of the proposed approach.

Some future works are to implement a more robust MI estimator for missing values, and to do an extensive comparison with other machine learning methods.

References

[1] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data.* John Wiley & Sons, New Jersey, USA, 2nd edition, 2002.

[2] O. Troyanskaya, M. Cantor, O. Alter, G. Sherlock, P. Brown, D. Botstein, R. Tibshirani, T. Hastie, and R. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[3] G. E. Batista and M. C. Monard. An analysis of four missing data treatment methods for supervised learning. *Appl. Artif. Intell.*, 17(5-6):519–533, 2003.

[4] N. Kwak and C.-H. Choi. Input feature selection by mutual information based on parzen window. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 24(12):1667–1671, 2002.

[5] F. Rossi, A. Lendasse, D. Francois, V. Wertz, and M. Verleysen. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometr. Intell. Lab. Syst.*, 80:215–226, 2006.

[6] T. Denoux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Trans. Syst. Man. Cybern. C Appl Rev*, 25(5):804–813, 1995.