Frequency Ratio: a method for dealing with missing values within nearest neighbour search

Rosanne Janssen[1], Pieter Spronck[2], Pauline Dibbets[1], Arnoud Arntz[3]

[1]Maastricht University, [2]Tilburg University, [3]University of Amsterdam, The Netherlands

rosanne.janssen@maastrichtuniversity.nl, p.spronck@tilburguniversity.edu, pauline.dibbets@maastrichtuniversity.nl, A.R.Arntz@UvA.nl

Abstract: *In this paper we introduce the Frequency Ratio (FR) method for dealing with missing values within nearest neighbour search. We test the FR method on known medical datasets from the UCI machine learning repository. We compare the accuracy of the FR method with five commonly used methods (three "imputation" and two "bypassing" methods) for dealing with values that are "missing completely at random" (MCAR) for the purpose of classification. We discovered that in most cases, the FR method outperforms the other methods. We conclude that the FR method is a strong addition to the commonly used methods for dealing with missing values within the nearest neighbour method.*

Key words: Frequency Ratio, missing values, nearest neighbour search, classification, MCAR

1. Introduction

Classification is used in a large variety of domains to gain insight in the structure of and relationships within datasets. Real-life datasets often have values missing. Most classification methods do not incorporate a way to deal with missing values. Instead, the user is expected to ignore cases with missing values or to somehow add the values that are missing.

In our research, we are particularly interested in datasets from the medical domain. In practice, such datasets suffer from considerable amounts of missing values. Values may be missing because of diverse causes, such as equipment malfunctions, lack of patient cooperation, or a lack of time to collect all information. Eliminating cases with missing values means ignoring potentially valuable information, which could lead to a less precise classification. Therefore elimination is risky as it might lead to invalid models, which might even have unethical and clinically unacceptable consequences.

For building a knowledge-based system or decision support system in the medical domain, classification is an often-used approach. Nearest neighbour search is commonly used to implement such classification, as it is widely known and easy to use. By itself, nearest neighbour search cannot deal with missing values. In practice, when implementing a nearest neighbour search algorithm for a dataset with missing values, the missing values are either replaced with an estimated value, or the developers try to bypass the missing values. Both approaches are problematic, as replacing generally assumes that one particular value should be used in place of the missing value, and bypassing may ignore valuable information.

In this study we introduce a novel method for dealing with missing values for the nearest neighbour search algorithm, which we call Frequency Ratio (FR). We present the FR method, and evaluate it by comparing its results on multiple datasets with the results of five commonly-used methods.

2. Missing values in data

This section discusses how missing values in data are dealt with in practice (2.1) and makes a distinction in types of missing values (2.2).

2.1 Dealing with missing values

Statistical literature has introduced many techniques for dealing with missing values within statistical analysis (Little and Rubin 1987, Allison 2002). Some researchers use nearest neighbour search to estimate missing values, which they call "nearest neighbour imputation" (Wasito and Mirkin 2005, Todeschini 1990, Malarvizhi and Thanamani 2012). Only a few researchers have investigated

methods for dealing with missing values in nearest neighbour search itself (Juhola and Laurikkala 2011).

Case deletion, i.e., the removal of cases that cannot be processed, is the most commonly proposed option for dealing with missing values in nearest neighbour search (Little and Rubin 1987, Juhola and Laurikkala 2011). Case deletion is often considered 'unethical', especially where medical datasets are concerned. Moreover, case deletion may remove useful information. As an alternative, imputation (i.e., replacement of missing values) is a popular technique, especially for clinical datasets (Juhola and Laurikkala 2011). The advantage of imputation is that it is simple and fast. Two commonly used methods of imputation are mode and mean imputation (see 3.2). Mode and mean imputation have been compared with nearest neighbour imputation (Batista and Monard 2003, Acuña and Rodriguez 2004). These studies found that nearest neighbour imputation outperformed mode and mean imputation. However, the influence of missing values on nearest neighbour classification results has seldom been investigated. Juhola and Laurikkala (2011) studied the effect of multiple rates of missing values on classification using nearest neighbour search. They found that the performance of the Heterogeneous Euclidean-Overlap Metric (HEOM; see 3.2) was equal to mean/mode imputation in general; only for one specific dataset mean/mode imputation performed better.

Mean and mode imputation are examples of single unconditional imputation. A more sophisticated approach is conditional imputation. An example is conditional mean imputation, which replaces a missing value with a mean that is estimated from the specific subgroup to which the subject with missing values belongs (Van der Heijden et al. 2006). It models the missing values as a function (e.g., regression) of the observed values (Conrady and Jouffe 2011). A more advanced form of conditional imputation is multiple imputation, in which multiple datasets are created with different imputations based on a random draw from estimated underlying distributions (Donders et al. 2006). After each complete dataset is analysed independently, estimates of parameters of interest are averaged to result in a single estimate (Royston 2004).

In the present study we investigate a novel method to deal with missing values within nearest neighbour classification, in particular when used in real-life knowledge-based or decision support systems that have to deal with missing values in real time. Such systems tend to work with datasets which are changing constantly and rapidly. Therefore, a method that deals with missing values must have relatively low computational costs and should be useable without human interaction. Methods like conditional and multiple imputation are therefore less suitable for such real-life systems.

2.2 Distinction in types of missing values

Three different types of missing values can be distinguished (Rubin 1976, Little and Rubin 1987, Allison 2002, Kaiser 2014):

1. Missing Completely at Random (MCAR): the probability of missing data on a particular attribute Y is not related to the value of Y itself nor to the values of any other attributes in the dataset; for example, MCAR may occur when part of the data is missing due to the failure of equipment.
2. Missing at Random (MAR): the probability of missing data on a particular attribute Y is not related to the value of Y, after controlling for other attributes in the dataset; for example, MAR may occur when participants refuse to specify their income because they consider it private information, but the missing value can be estimated (except for a random error) on the basis of known variables.
3. Non-ignorable (NI; also known as Not Missing at Random (NMAR)): the probability of missing data on a particular attribute Y may depend on the values of the attribute; for example, NI may occur when some participants refuse to specify their income out of embarrassment, as they consider it relatively low.

How to deal with missing values depends on the type. In this paper we focus specifically on the MCAR type.

3. Nearest neighbour search

This section gives an introduction to the nearest neighbour search method (3.1) and discusses the commonly methods for dealing with missing values within nearest neighbour search (3.2).

3.1 Nearest neighbour search method

Nearest neighbour search is one of the best-known search algorithms. It classifies an instance by assigning it the most common class of a specific number of its nearest neighbours. Several variants of defining neighbourhood are used. The most common one is the Euclidian Distance function, which is defined in Formula 1 (Juhola and Laurikkala 2011, Wilson and Martinez 1997):

(1) $$D(x,y) = \sqrt{\sum_{i=1}^{m} d(x_i,y_i)^2}$$

in which *x* and *y* are two instances and *i* is an attribute.

For nominal (numerical values with no order in rank) attribute types the distance value $d(x_i,y_i)$ is defined as:

$$d(x_i,y_i) = \begin{cases} 1, x_i \neq y_i \\ 0, x_i = y_i \end{cases}$$

For ordinal (a finite number of numerical values with an order in rank) attribute types the distance value $d(x_i,y_i)$ is defined by:

$$d(x_i,y_i) = \frac{|x_i - y_i|}{R_i}$$

in which R_i is the range of attribute *i* used to normalize the absolute distance.

Nearest neighbour search is a search algorithm that seeks "similar" instances. Often the number of instances to be found is a fixed number. In that case, we call it *k*-nearest neighbour search (kNN), where *k* is the number of instances to be found. 1-Nearest neighbour search (1-kNN) only finds the one instance to which the distance is the smallest. It classifies the new instance by assigning the class of the found instance.

The advantages for classification with kNN are that training is very fast, that it is simple and easy to learn, and that it is robust to noisy training data (Bhatia and Vandana 2010). A disadvantage is that kNN is biased by the value of *k*. It also has a high computational complexity and is therefore limited by the amount of available memory. Moreover, kNN is easily misled by irrelevant attributes, which may be dealt with by using feature weights (Bhatia and Vandana 2010). A weakness of the nearest neighbour search method is that, in principle, it cannot be used when the dataset contains missing values.

3.2 Commonly methods for dealing with missing values within kNN

The most commonly used methods for dealing with MCAR missing values within nearest neighbour search can be divided into two types:

1. *Imputation*: The missing value x_i for an attribute *i* in Formula 1 is replaced. After replacing x_i the formula can be used normally. The value which replaces x_i is commonly based on existing values for that attribute in the dataset. This means that imputation makes use of domain knowledge.
2. *Bypassing*[1]: The function for the distance value $d(x_i,y_i)$ in Formula 1 is eliminated or replaced with a fixed value. I.e., the distance value is not depended on attribute *i*. After replacing $d(x_i,y_i)$ the formula can be used normally. The value which replaces $d(x_i,y_i)$ does not depend on other instances in the dataset. This means that bypassing does not make use of domain knowledge.

The most commonly used imputation methods within nearest neighbour search are:

- Mode imputation (MOI): this method replaces the missing value of x_i with the mode of the existing values for the corresponding attribute *i* in the same dataset (Batista and Monard 2003).
- Mean imputation (MI): this method replaces the missing value of x_i with the mean of the existing values for the corresponding attribute *i* if the attribute is ordinal, and with the mode if the attribute is nominal (Batista and Monard 2003, Folleco, Khoshgoftaar and Hulse 2008). This type of imputation is also called "unconditional mean imputation."

[1] The term "bypassing" is not found as such in the literature. We introduced the term to refer to methods that replace ("bypass") the distance value when missing values are concerned.

- Median imputation (MDI): this method replaces the missing value of x_i with the median of the existing values for the corresponding attribute *i* if the attribute is ordinal, and with the mode if the attribute is nominal. Median imputation is used instead of mean imputation when the mean may be affected by the presence of outliers (Acuña and Rodriguez 2004).

The most commonly used bypassing methods within nearest neighbour search are:

- Ignore method (IGN): this method uses Formula 1 but ignores the attributes that have a missing value in at least one of the two instances. To compare the distance values $D(x,y)$ between different pairs of instances, the distance values must be normalized, because the comparison of other pairs of instances may be based on different attributes. Therefore, the distance values are divided by the square root of the number of attributes that contribute to the distance measure (i.e., the attributes that have a value in both instances) (Aha 1990). IGN is defined in Formula 2 (Aha 1990) as a variation of Formula 1:

(2) $$D(x,y) = \frac{\sqrt{\sum_{i=1}^{m} d(x_i,y_i)^2 \times Both\_known\,(x_i,y_i)}}{\sqrt{\sum_{i=1}^{m} Both\_known\,(x_i,y_i)}}$$

where

$$Both\_known(x_i,y_i) = \begin{cases} 1, if\ x_i and\ y_i\ both\ have\ values \\ 0, otherwise \end{cases}$$

- Heterogeneous Euclidean-Overlap Metric (HEOM): this method assumes that attributes have a maximum difference in a pairwise attribute-value comparison if one of the values is missing (Aha 1990). HEOM is defined in Formula 3 (Juhola and Laurikkala 2011) as a variation of Formula 1:

(3) $$D(x,y) = \sqrt{\sum_{i=1}^{m} d(x_i,y_i)^2}$$

The variation is reflected in the definition of $d(x_i,y_i)$. For nominal (numerical values with no order in rank) attribute types the distance value $d(x_i,y_i)$ is defined by:

$$d(x_i,y_i) = \begin{cases} 1, x_i \neq y_i \\ 1, x_i\ or\ y_i\ missing \\ 0, x_i = y_i \end{cases}$$

and for ordinal (a finite number of numerical values with an order in rank) attribute types the distance value $d(x_i,y_i)$ is defined by:

$$d(x_i,y_i) = \begin{cases} \frac{|x_i - y_i|}{R_i} \\ 1, x_i\ or\ y_i\ missing \end{cases}$$

4. Frequency Ratio (FR)

As an alternative for the commonly used methods that deal with missing values in nearest neighbour search discussed above, we introduce Frequency Ratio (FR). FR is a method that replaces the distance value $d(x_i,y_i)$ in Formula 1 with a value based on a probability rate of the missing value x_i and all the possible values of y_i in the domain.

We always have a target instance Y that we want to classify and more instances X we want to compare Y to in order to achieve the classification. For this comparison we ignore all the attributes that are missing in Y, because they are missing in every comparison where we compare Y with an instance X.

FR is handled differently for nominal attributes and ordinal attributes. We explain FR for nominal attributes in Subsection 4.1, and FR for ordinal attributes in Subsection 4.2. In Subsection 4.3 we discuss FR's expected performance.

4.1 FR for nominal attributes

For nominal attributes $d(x_i,y_i)$ Formula 1 is replaced by Formula 4 if x_i is missing.

(4) $$d(x_i, y_i) = \frac{\sum_{z=1}^{n} FR\_nom(z_i y_i)}{\sum_{z=1}^{n} not\_missing(z_i)}$$

$d(x_i,y_i)$ is based on all n instances in the dataset; z_i is the value of attribute i for instance z

with* $$FR_{nom(z_i,y_i)} = \begin{cases} 0, z_i = y_i \\ 1, z_i \neq y_i \\ 0, z_i \; is \; missing \end{cases}$$

and $$not\_missing(z_i) = \begin{cases} 1, if \; z_i \; is \; not \; missing \\ 0, if \; z_i \; is \; missing \end{cases}$$

* Because we ignore all the attributes that are missing in Y, y_i *is missing* is not an option.

Intuitively, this means that for nominal attributes, if an attribute is missing it is replaced by a fraction that represents the probability that the attribute is 1. E.g., in a dataset where 15% of the instances have a value of 1 for attribute *i*, 60% have a value of 0 for attribute *i*, and the remaining 25% have attribute i missing, the value of 0.20 is used for attribute *i* when it is missing.

4.2 FR for ordinal attributes

For the ordinal attributes $d(x_i,y_i)$ is also replaced by a value that depends on the distribution of the values of attribute *i* in the whole dataset. Because for ordinal types there is an order in rank, we can take the values for attribute *i* in the whole dataset that are equal to y_i into account, but also the values that are in close proximity to y_i. We may vary the effect of proximity to y_i by using a proximity parameter *a*.

For the ordinal attributes $d(x_i,y_i)$ Formula 1 is replaced by Formula 5 if x_i is missing.

(5) $$d(x_i, y_i) = \frac{\sum_{z=1}^{n} FR\_ORD(z_i, y_i, a)}{\sum_{z=1}^{n} not\_missing(z_i)}$$

with $$FR\_ORD(z_i, y_i, a) = \begin{cases} 0, if \; (y_i - a \times R_i) \leq z_i \leq (y_i + a \times R_i) \\ 1, otherwise \end{cases}$$

and $$not\_missing(z_i) = \begin{cases} 1, if \; z_i \; is \; not \; missing \\ 0, if \; z_i \; is \; missing \end{cases}$$

$d(x_i,y_i)$ is based on all n instances in the dataset; z_i is the value of attribute i for instance z; a is a fraction between 0 and 1; and R_i is the difference between the highest and lowest values of attribute i in the dataset.

Intuitively, this means that for ordinal attributes, the distance between two instances X and Y for an attribute *i* that is missing in X, is 1 minus the chance that X is the same (or close to the) value for attribute *i* as Y, based on the distribution of *i* in the whole dataset.

4.3 Expected performance of FR

FR is a mix between imputation and bypassing. It replaces the distance value in a way similar to bypassing methods, but, like imputation and unlike common bypassing methods, makes use of domain knowledge. The domain knowledge that FR uses is information about the distribution of the instances in the dataset of the missing attribute. FR translates this distribution into a probability rate.

Compared to bypassing methods we expect a higher accuracy rate with FR because FR makes use of domain knowledge to replace $d(x_i,y_i)$. Compared to imputation methods we expect a higher accuracy rate with FR because imputation methods only replace x_i with a fixed value while FR replaces the outcome of the distance function that is based on x_i with a value that is based on probability. A consequence is that for FR the distance function $d(x_i,y_i)$ can only have the value 0 when all the instances in the dataset have the same value for an attribute. In this, it differs from other imputation methods, in particular from mode imputation.

The commonly used imputation methods within the nearest neighbour search are single imputation methods. Single imputation methods generally underestimate the standard error of estimates. A single imputation pretends that we know the unobserved value with certainty, when actually it is unknown but estimated by the imputation method (He 2000). This is also applicable to FR, because it is inherent in the method. Nevertheless, we argue that FR is more reliable than the commonly used imputation methods, by means of the following example.

Assume that we have a dataset with 50 instances, with for attribute *x* a distribution of values as shown in Figure 1. The values for the mean, mode and median are 5.56, 6 and 1 respectively. Imputation with the mean or mode is not representative for the dataset, as there are no instances where *x*=5 and only two instances where *x*=6. Imputation with the mode of 1 poses the problem that the distance to an attribute with *x*=1 is 0, but the distance to an attribute with *x*=9 is the maximum 1. This is not desirable as in this dataset the number of instances with *x*=1 and the number of instances with *x*=9 are almost equal.

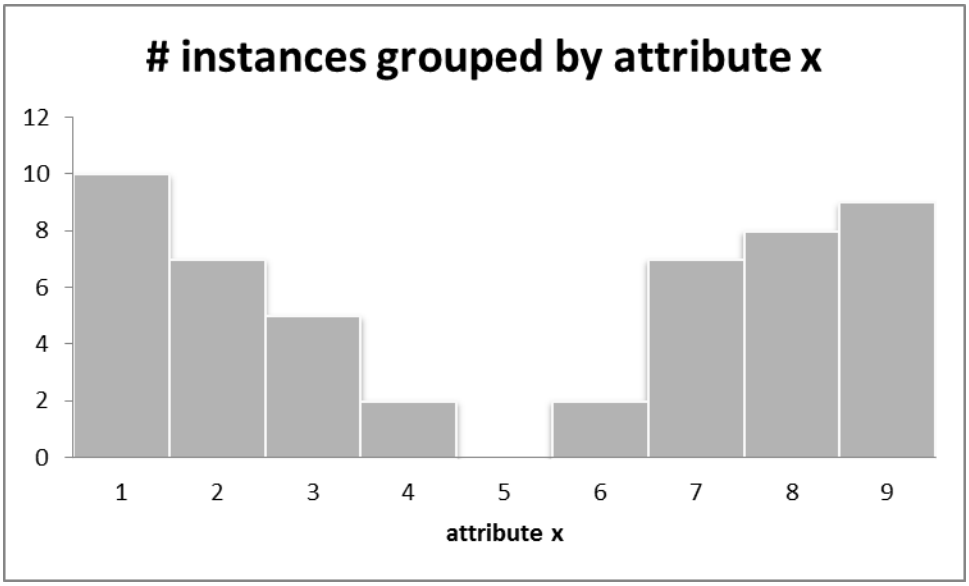

Figure 1: Distribution of the instances in example dataset

FR solves this problem by taking the distribution into account. It replaces the distance function with a probability rate based on the distribution of the instances over the values of the attribute. FR provides a distance that is more consistent with the reality. The probability that a missing value for *x* originally had the value 5 is nil. So a comparison with an instance with *x*=5 should give a large distance. In this case FR gives the maximum distance 1.

Finally, we want to comment on computational costs, which tend to be high for nearest neighbour search. For *n* instances and *m* attributes they are O(*nm*). When FR is applied to deal with missing values, the distance function is replaced by one of the same order of complexity. Therefore, FR does not increase computational costs.

5. Experimental design

We compared FR to five different commonly used methods for dealing with missing values in nearest neighbour search. For these experiments, Subsection 5.1 describes the used datasets, Subsection 5.2 describes how we prepared the datasets, and Subsection 5.3 explains the experimental setup.

5.1 Datasets

For this experiment we used 5 datasets from the UCI machine learning repository (Frank and Asuncion 2010), namely Liver Disorders Data Set (BUPA), Haberman's Survival Data Set (Haberman), Breast Cancer Wisconsin Original (BCW), Acute Inflammations Data Set (AI), and SPECT Heart Data Set (SPECT). In **Table 1** the datasets are specified.

Table 1: Datasets used in experiment

Dataset	#Instances	#Nominal attr.	#Ordinal attr.	Classes	Missing values	Accuracy 1-kNN
BUPA	345	0	6	1 (42%) 2 (58%)	no	62.6 %
Haberman	306	0	3	1 (72%) 2 (28%)	no	65.0 %
BCW	699	0	9	1 (66%) 2 (34%)	yes (16 values are missing)	95.0 %
AI-D1	120	5	1	1 (51%) 2 (49%)	no	100 %
AI-D2	120	5	1	1 (58%) 2 (42%)	no	100 %
SPECT	267	22	0	1 (21%) 2 (79%)	no	76.8 %

All the nominal attributes in the datasets are binominal, with only the values 0 and 1. The AI dataset contains two unrelated class attributes, D1 and D2. The dataset with the 6 attributes can be used to classify D1 or D2. We refer to this dataset with two separate class attributes as to two different datasets, AI-D1 and AI-D2. In the BCW dataset there are 16 values missing. This is less than 0.003% of the total of values in the dataset.

5.2 Dataset preparation

For every dataset we artificially created new datasets with missing values, by randomly marking values within instances as missing. This way we created 100 datasets of each for each of the following percentages of missing data: 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 20%, 30%, 40%, and 50% (1400 datasets in total). Values missing in any of the sets are not necessarily missing in any of the other sets. We did not include datasets with more than 50% missing values. Juhola and Laurikkala (2011) found good results for classification in two-class datasets up to 20-30% missing values, but showed that it is not sensible to use datasets with higher percentages of missing values.

5.3 Setup

For the experiment[2] we applied 1-kNN search for classification using Formula 1 in Subsection 3.1. For classifying an instance Y we only used the attributes in the nearest neighbour search that were present in Y. Therefore the only missing attributes were in the instances we compared Y to.

To compare FR with the five commonly used methods for dealing with missing values (see 3.2) we tested the method on each dataset for classification, using 'leave-one-out cross validation'. Witten et al. (2011) indicates that this approach achieved the closest possible accuracy estimation for datasets of the size we used. The accuracy for the classification is defined by formula 6:

(6) $$accuracy = \frac{r}{n} \times 100\%$$

in which *r* is equal to the number of correctly classified instances and *n* the number of all instances (with at least one attribute that is not missing). Table 1 (rightmost column) shows the accuracy for the original dataset.

For each of the six datasets, we compare results of 1-kNN using FR and each of the five commonly used methods, for 14 different percentages of missing values. As we have 100 different variations of the dataset for each of these comparisons, we report the mean accuracy achieved over these 100

[2] *Source code is available on request from the first author.*

variations. We chose to use kNN in its pure form, without weighting the attributes, because the choice of which weighting method is used has an impact on the accuracy of the nearest neighbour search.

6. Results

Before FR can be used, a value must be decided upon for the proximity parameter *a* in Formula 5, which we do in Subsection 6.1. We then compare FR with the commonly used methods in Subsection 6.2.

6.1 Choice of FR variant

To decide which variant of FR for ordinal types seems to offer the best performance, we ran our experiment as described in Section 5 for different values of parameter *a*, namely 0.05, 0.10, 0.15, 0.20 and 0.25. We call these variants of FR: FR5, FR10, FR15, FR20 and FR25 respectively. We only ran the experiment on the datasets BUPA, Haberman and BCW, as SPECT does not have ordinal attributes and AI-D1 and AI-D2 only have one ordinal attribute. We ran the experiment with 1-kNN on all the created datasets with missing values using FR5, FR10, FR15, FR20 and FR25. This allowed us to calculate the accuracy for every percentage of missing values for every dataset per variant of FR. The results gave us an indication that FR20 is the best variant for the datasets that we tested, with FR15 as a close second. We therefore decided to use FR20 for the next set of experiments.

6.2 FR compared with commonly used methods

After we decided on FR20 to be the preferred variant, we compared FR20 with the five commonly used methods discussed in Subsection 3.2. Figures 2 to 6 show the results of t-tests for these comparisons. The left graphs shows the results for the datasets with percentages of missing values from 1% to 9%, and the right graphs show the results for the datasets with percentages of missing values from 10% to 50%. The x-axis of each of the graphs shows the percentage of missing values in the dataset. The y-axis shows the effect size *r* of the t-test (using Formula 7, in which *t* refers to the outcome of the t-test and *df* refers to the degree of freedom) for the means of accuracy for the specific combination of FR20 and the compared method. The lines p1 and p2 mark the areas of a significant difference. Above p2 FR20 is significantly better than the compared method. Below p1 the compared method is significantly better than FR20. Between the lines FR20 does not differ significantly from the compared method.

(7) $$r = \begin{cases} \sqrt{t^2/(t^2+df)}, t \geq 0 \\ -\sqrt{t^2/(t^2+df)}, t < 0 \end{cases}$$

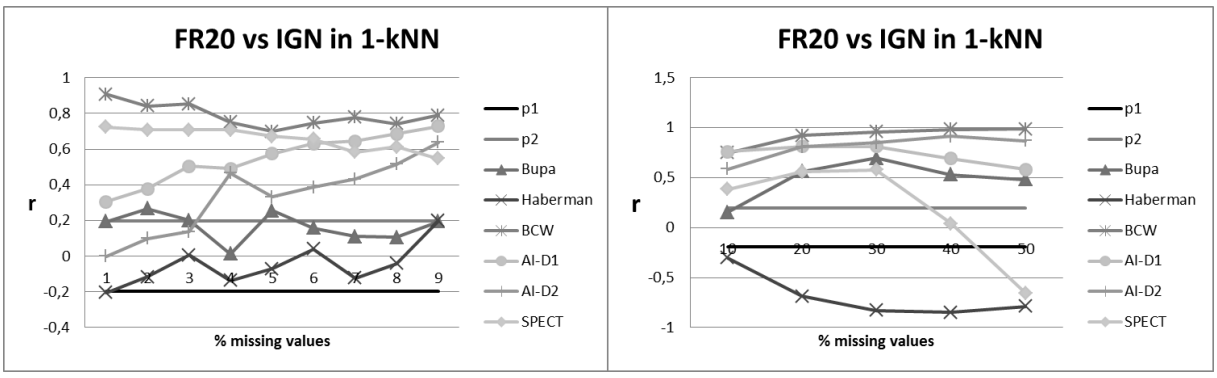

Figure 2: Comparison between FR20 and IGN

When comparing FR20 with IGN (Figure 2) we note that FR20 is always significantly better or that there is no significant difference with IGN for the percentages of missing values below 10%. Only on the Haberman dataset FR20 is never significantly better (but neither is it significantly worse). For the percentages of 10% missing values and higher we see the same results, except for Haberman (IGN is significantly better) and SPECT (IGN is significantly better for 50%).

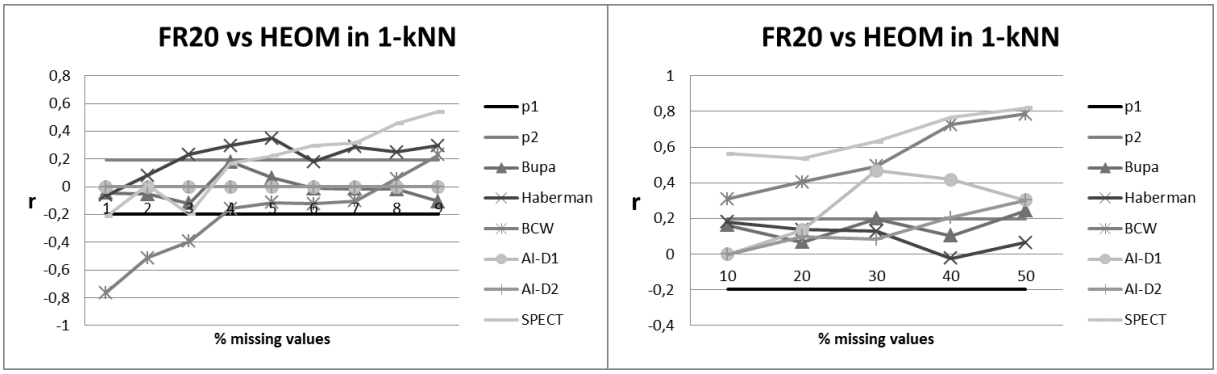

Figure 3: Comparison between FR20 and HEOM

When comparing FR20 with HEOM (Figure 3) we note that FR20 is always significantly better or that there is no significant difference with HEOM, except for BCW. For BCW HEOM is significantly better than FR20 when there are 3% or less of the values missing, but FR20 is significantly better than HEOM when there are 9% or more of the values missing. The difference of BCW with the other datasets is that BCW has the largest number of instances. Also, for BCW each attribute has a dominant value.

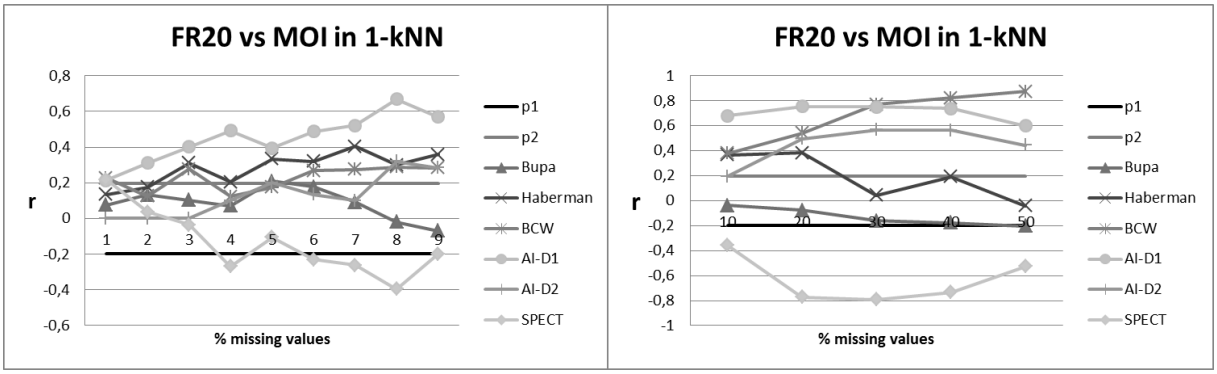

Figure 4: Comparison between FR20 and MOI

When comparing FR20 with MOI(Figure 4) we note that FR20 is always significantly better or that there is no significant difference with MOI, except for the dataset SPECT. For SPECT MOI is significantly better for 4% and 6% and more missing values.

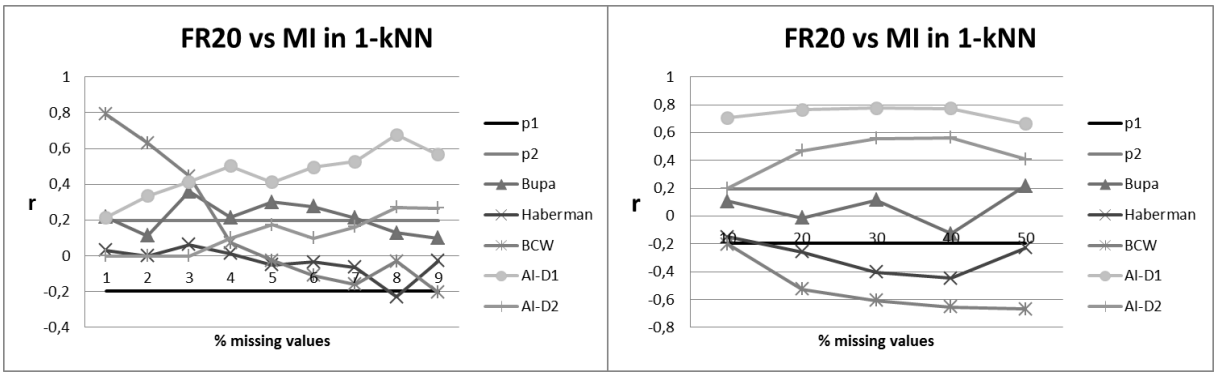

Figure 5: Comparison between FR20 and MI

When comparing FR20 with MI (Figure 5; SPECT is not shown as it only has nominal attributes) we note that FR20 is always significantly better or that there is no significant difference with MI for the percentages of missing values below 10% (except for the Haberman dataset with 8% missing values). For the Haberman dataset and the BCW dataset FR20 is never significantly better than MI.

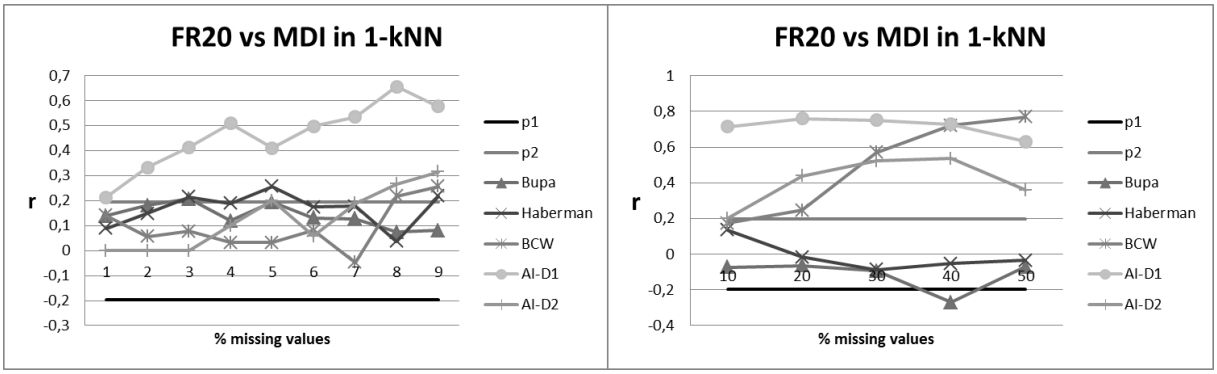

Figure 6: Comparison between FR20 and MDI

When comparing FR20 with MDI (Figure 6; SPECT is not shown as it only has nominal attributes) we note that FR20 is always significantly better than MDI or there is no significant difference with MDI, except for the BUPA dataset with 40% missing values.

7. Discussion

We start this section by explaining the deviant results on the Haberman dataset in the comparisons with IGN and MI (10% or more missing values) and the SPECT dataset in the comparison with MOI.

We included the Haberman dataset because it is a medical dataset, which was used in another study that investigated dealing with missing values within the nearest neighbour method (Juhola and Laurikkala 2011). After analysing our results on the Haberman dataset we found that it has a dominant outcome class with 72% of the instances within one class, i.e., it has a baseline accuracy of 72%. The accuracy for the Haberman dataset without missing values were below the baseline, which is an indication that nearest neighbour search simply does not work well on this dataset. When missing values are introduced, the accuracy actually improves, approaching the baseline accuracy of 72%. The same problem occurs within the SPECT dataset where the accuracy for the dataset without missing values is lower (77%) than the baseline (79%). Therefore, we believe that the nearest neighbour search simply is unsuitable to work with them, regardless of what method is used to deal with missing values.

Comparing the accuracy of the results of 1-kNN using FR20 to deal with missing values on the remaining datasets, to the results achieved with the five commonly used methods IGN, HEOM, MOI, MI, and MDI, the accuracy achieved by FR20 tends to be equal or significantly higher. This holds in particular for datasets with nominal attributes. An exception should be made for the BCW dataset. HEOM is significantly more accurate for percentages of missing values of 3% or less, but for percentages of 9% of or more missing FR20 is significantly more accurate than HEOM for the BCW dataset. For MI we found the opposite pattern for the BCW dataset. For the BCW dataset FR20 is significantly more accurate than MI for 3% of missing values of less. For higher percentages of missing values MI is significantly more accurate than FR20. This variety in results for the BCW dataset may have been caused by the high diversity in this dataset and the large number of instances that it contains, combined with 1-kNN which selects only a single neighbour to compare to. In general, for larger datasets higher values for *k* work better.

Beside the experiments reported in this paper, we also ran all the experiments for 5-kNN. The results for 5-kNN were very similar to these reported with 1-kNN.

In this study we analysed the accuracy of the FR method on datasets where the missing values are “missing completely at random” (MCAR), and found that it performs better than the commonly used methods on the datasets that we used. As it is not possible to create datasets with values “missing at random” (MAR) without substantial knowledge of the exact contents of the dataset, we cannot claim the same for MAR values. However, because for both MCAR and MAR the probability of missing data on an attribute Y does not depend on the values of that attribute, we expect that our results would hold for datasets with missing values of the type MAR. To check this assumption further research needs to be done. Whether or not FR performs better or worse than any of the commonly used methods for dealing with missing values of the type NMAR, depends on the underlying reasons why the values are missing – however, it is certain that if those underlying reasons are known, a method can be devised

for the dataset that is superior to FR and all of the commonly used methods, namely a method that estimates the missing value based on the dependencies.

8. Conclusion

In this paper we introduce the Frequency Ratio (FR) method for dealing with missing values of the type MCAR in nearest neighbour search. FR replaces the distance value in the Euclidian distance function used by the nearest neighbour method for missing values while making use of domain knowledge, namely the distribution of values for the corresponding attribute in the dataset. To evaluate FR we compared it with five commonly used methods, namely Ignore (IGN), Heterogeneous Euclidian Overlap Method (HEOM), Mode Imputation (MOI), Mean Imputation (MI), and Median Imputation (MDI). Our test set consisted of six datasets from within the medical domain, from which we randomly removed different percentages of values.

We compared the performance of FR20 (FR with its proximity parameter set to 20%) with the five commonly used methods, for the different percentages of missing values. These comparisons demonstrated that in almost all cases, the accuracy of the results achieved with FR20 is significantly higher than, or not significantly different from, each of the five alternative methods.

We conclude that FR20 may be expected to perform better than the commonly used methods for dealing with missing values in nearest neighbour search with values “missing completely at random” (MCAR). A promising feature is that this can be extrapolated to missing values of the type MAR. We will investigate this extrapolation in future research.

Bibliography

Acuña, E. & C. Rodriguez, 2004: The Treatment of Missing Values and its Effect on Classifier Accuracy. In *Classification, Clustering, and Data Mining Applications*, 639-647. Springer Berlin Heidelberg

Aha, D. W., 1990: A study of instance-based algorithms for supervised learning tasks. Irvine, CA: University of California

Allison, P. D., 2002: *Missing data.* Thousand Oaks: Sage Publications

Batista, G. E. A. P. A. & M. C. Monard, 2003: An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence,* 17**,** 519-533

Bhatia, N. & Vandana, 2010: Survey of Nearest Neighbor Techniques. *International Journal of Computer Science and Information Security,* 8**,** 302-305

Conrady, S. & L. Jouffe, 2011: Missing Values Imputation: A New Approach to Missing Values Processing with Bayesian Networks. 35. Conrady Applied Science

Donders, A. R. T., G. J. M. G. van der Heijden, T. Stijnen & K. G. M. Moons, 2006: Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology,* 59**,** 1087-1091

Folleco, A., T. M. Khoshgoftaar & J. Hulse, 2008: Software Fault Imputation in Noisy and Incomplete Measurement Data. In *Recent Advances in Reliability and Quality in Design,* ed. H. Pham, 255-274. Springer London

Frank, A. & A. Asuncion, 2010: UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science

He, Y., 2000: Missing Data Analysis Using Multiple Imputation: Getting to the Heart of the Matter. *Circ Cardiovasc Qual Outcomes,* 3**,** 98-105

Juhola, M. & J. Laurikkala, 2011: Missing values: how many can they be to preserve classification reliability? *Artificial Intelligence Review***,** 1-15

Kaiser, J., 2014: Dealing with Missing Values in Data. *Journal of Systems Integration,* 1**,** 42-51

Little, R. J. A. & D. B. Rubin, 1987: *Statistical analysis with missing data*. New York: Wiley

Malarvizhi, R. & A. S. Thanamani, 2012: K-Nearest Neighbor in Missing Data Imputation. *International Journal of Engineering Research and Development,* 5**,** 5-7

Royston, P., 2004: Multiple imputation of missing values. *The Stata Journal,* 4**,** 227-241

Rubin, D. B., 1976: Inference and missing data. *Biometrika,* 63**,** 581-592

Todeschini, R., 1990: Weighted k-Nearest Neighbour Method for the Calculation of Missing Values. *Chemometrics and Intelligent Laboratory Systems,* 9**,** 201-205

Van der Heijden, G. J. M. G., A. R. T. Donders, T. Stijnen & K. G. M. Moons, 2006: Imputation of missing values is superior to complete case analysis and the missing-indicator method in multivariable diagnostic research: A clinical example. *Journal of Clinical Epidemiology,* 59**,** 1102-1109

Wasito, I. & B. Mirkin, 2005: Nearest neighbour approach in the least-squares data imputation algorithms. *Information Sciences,* 169**,** 1-25

Wilson, D. R. & T. R. Martinez, 1997: Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research,* 6**,** 1-34

Witten, I. H., E. Frank & M. A. Hall, 2011: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc.

JEL Classification: C45, C82