ORIGINAL RESEARCH

Diagnostic with incomplete nominal/discrete data

Herbert F. Jelinek *[1,2], Andrew Yatsko [1], Andrew Stranieri [1], Sitalakshmi Venkatraman [3], Adil Bagirov [1]

[1] Centre for Informatics and Applied Optimisation, Federation University, Ballarat, VIC, Australia
[2] Centre for Research in Complex Systems and School of Community Health, Charles Sturt University, Albury, NSW, Australia
[3] Department of HE-Business (IT), Melbourne Polytechnic, Prahran, VIC, Australia

Received: November 19, 2014 **Accepted:** December 16, 2014 **Online Published:** January 15, 2015
DOI: 10.5430/air.v4n1p22 **URL:** http://dx.doi.org/10.5430/air.v4n1p22

Abstract

Missing values may be present in data without undermining its use for diagnostic / classification purposes but compromise application of readily available software. Surrogate entries can remedy the situation, although the outcome is generally unknown. Discretization of continuous attributes renders all data nominal and is helpful in dealing with missing values; particularly, no special handling is required for different attribute types. A number of classifiers exist or can be reformulated for this representation. Some classifiers can be reinvented as data completion methods. In this work the Decision Tree, Nearest Neighbour, and Naive Bayesian methods are demonstrated to have the required aptness. An approach is implemented whereby the entered missing values are not necessarily a close match of the true data; however, they intend to cause the least hindrance for classification. The proposed techniques find their application particularly in medical diagnostics. Where clinical data represents a number of related conditions, taking Cartesian product of class values of the underlying sub-problems allows narrowing down of the selection of missing value substitutes. Real-world data examples, some publically available, are enlisted for testing. The proposed and benchmark methods are compared by classifying the data before and after missing value imputation, indicating a significant improvement.

Key Words: Classification, Missing values, Categorical data, Continuous features, Discretization

1 Introduction

Missing entries may appear in database records for many reasons. Take, for example, the aspect of timeliness - data is obtainable but not yet available; or consider the acquisition costs - data is generally valuable but can be dealt without. Above all, there are case specifics - not all of the data is required or some of it is unobtainable. This situation is not only very common, but actually justifies the presence of Missing Values (MVs) - data has all it needs, if someone is only able to embrace its logic.

It is acceptable and, in fact, unavoidable to have MVs, especially in clinical practice, but presents a problem for outright classification which is data based. Yet, hypothetically, these values may be requisitioned. Even if values are illogical in their positions on a particular record, appropriate entries can be made to acknowledge this fact. In practical terms, MVs can be substituted with some actual values from ranges pertaining to their attributes. For example, it is routinely suggested that numerical MVs can be entered by their attribute mean. Even though this may distort data distribution and cause class noise, the genuine bulk of the data should be able to absorb the stress. However, any value or a combination of values will do well for this reason, perhaps random ones even better. There is another wisdom in using means for numerical attributes or modes for categorical attributes,

*Correspondence: Herbert F. Jelinek; Email: hjelinek@csu.edu.au; Address: Centre for Research in Complex Systems and School of Community Health, Charles Sturt University, Albury, NSW 2640 Australia

and in relevant subsets sooner than all data – these estimates of essential moments of data distribution are preserved in MV substitution. Let us embark on this path, yet produce as little noise as possible.

MVs in numerical data create a representation difficulty in classification problems. Only if an attribute is, in fact, discrete, for example, integer, some value outside its domain can be used to denote MV. However, such an assignment of a dummy value cannot be carried out as elegantly with continuous features. For instance, insertion of dummy data (e.g. "999.999") invariably distorts distance metrics underpinning classifiers. Therefore, it is more appropriate and common practice to acknowledge MVs using binary ("yes/no") supporting attributes. This implies that MVs can then be arbitrarily entered. However, no data mining algorithm applied directly to the data will be able to make a connection between the principal and their supporting attributes. Conversely, no representation problem exists with all nominal / discrete attributes where the "missing" value can be entered as a designated category. Apart from universality of formulation, this provides an additional rationale to discretise continuous attributes in mixed attribute type domains. This methodology is embraced in this work.

Conversion of a data mining method whereby it can do without referencing MVs is sometimes possible but is unwarranted. At the same time, there is a need for techniques serving auxiliary purposes, such as comparison, many of which are implemented into a software. A pre-processing step which fits MVs out with surrogates offers a systematic approach to the problem under these circumstances. Simple deletion of attributes and instances containing MVs is hardly an alternative - the remaining information may be insufficient to train a classifier. In fact, having too many of attributes, instead of being helpful, is counterproductive for instance deletion, despite ostensibly providing more data. It is not difficult to imagine that this will cause a cumulative effect under the "missing completely at random" scenario,[1] even if only a small amount of information per attribute is missing.[2]

MVs are often substituted by their attribute mode or mean, depending on whether the attribute type is categorical or numerical. By restraining change of other involved variables, the mode or mean can be evaluated more specifically. In ref.[3] we followed an approach having likeness of the one known as the General Location Model[4] to set MVs in one of the data examples. We argued that in mixed attribute type domains categorical attributes naturally subdivide data into clusters, that is, subspaces defined by different categories have to be assumed infinitely removed from each other; and used a small number of well defined, regarded influential features of this type to set MVs in all other attributes by mean or mode for each combination of values of the selected attributes. Any MVs in the class attribute were set independently from a single strongest, valued predictor. Other criteria can also be applied when subdividing data. For example, a dataset can be restructured into smaller sets, so as to minimize the impact of MVs. Latkowski and Mikolajczyk[5] decompose data into subsets based on statistical properties of the former. A restructuring approach that involves organising features into a hierarchy in consultation with experts was advanced by Stranieri and Zeleznikow.[6]

Narrowing down the selection is a general method, but it does not guarantee that classification is not disadvantaged as the substitute values are subset dependent. However, this variation does not seem to be as important for weak features. Rahman and Islam[7] adapt the C4.5 algorithm from ref.[8] for MV imputation. In their method MVs are entered by classifying data with respect to any affected feature. The same scheme involving the Naïve Bayesian classifier is applied in ref.[9] This may seem controversial as high precision can only be expected if the class and supporting attributes are highly correlated, but this is not necessarily the case in diagnostics. In fact, for the purpose of determining parameters of an assumed model of data distribution, an analogous application of regressive imputation in the Expectation Maximisation (EM) and the Multiple Imputation (MI) techniques by Rubin and colleagues,[1] suggests using auxiliary variables for a closer projection of imputed values.[2] However, the question is also about how precise the projected values need to be. Nevertheless, the focus on the original classification problem is lost with this approach.

Different importance of selecting the correct value can be illustrated on a data where a feature aligns strongly with the class, so predictions can be made directly from values of that feature. For example, the pre-diabetic condition in non-diabetic patients can be assessed from the blood glucose level alone. If such a dataset had MVs in other attributes, arguably, they could be set arbitrary without consequences. However, if the classification were attempted without that feature, it might fail due to the added noise, if not for weakness of the reduced feature-set. Hence, the aspect of class noise is critical, and it has a broader impact if features are allowed to recombine. Proposed in the current work techniques can be placed fairly into the category of anti-noise measures, although not in an intrusive but evasive form. We discuss the intrusive measures in ref.[10] In that regard, using the paradigm of attribute space facilitates reckoning; however, applying the notion is not straightforward in all-nominal / discrete data domains. Techniques, akin developed in ref.[11] (Gamberger and Lavrač), should instead be considered. The idea there is to generate decision rules via inductive logic programming, similar to rules deducible from a decision tree, and verify their components instance-by-instance. Instances that misconstrue many such literals are then regarded foreign. With MVs in data, rules are not rigid, so they can be varied to accommodate any uncertain instances. Returning therefore to the opening argument, since an optimal decision tree relies on attributes

that strongly correlate with the class, the impact of MVs in these attributes should be examined in the first place. Following this principle, several data structures resulting from different classifier designs are exploited in the current work to guide the deduction of a subset from which MVs could be selected.

2 Related work

Junninen et al.[12] used linear and cubic spline interpolation in conjunction with neural networks for MV imputation in time series data. Generally, in a multivariate context a relationship existing between attributes can be exploited. Wang and Rao,[13] Zhang et al.[14] evaluated approaches using kernel imputation. Tseng et al.,[15] Zhang et al.[14] combined clustering and regression. More generally, a model of data can be fit to a sample even though in the presence of MVs, as in the EM and MI techniques previously mentioned.[1] Many statistical software packages implement EM and MI: SPSS, SAS, to name a few.[16] In using the software one should keep in mind that the data distribution is often assumed to be multivariate normal. If this is not the case, it is sometimes possible to impose a transformation to that effect, at least in respect of individual variables. An account of available software facilitated modelling using MI in diabetes studies is given in ref.[17]

Despite regarded as "state of the art", EM and MI techniques are computationally very intensive, especially MI, which is rather a statistical experiment featuring an imputation method. Apart from the design, the biggest contributor to the problem is the multitude of model parameters as their number is dependent on the number of problem dimensions and can grow explosively with model complexity.[2] Therefore, only models involving a modest number of variables can be realistically evaluated. The quest for using a large number of parameters has yet to be substantiated with the amount of data available; otherwise, there may be a dramatic loss of precision of statistical inference - a phenomenon known as the "curse of dimensionality".[18] Above all, complexity alone does not define the model quality. The model should also be descriptive of the data. This is inherently difficult to achieve because of a paradox in the formulation of the analytics exercise: on one hand, one cannot readily frame a question of interest without knowing what data is available, and on the other, one cannot identify what data is required without knowing the question of interest. The modelling requires a substantial amount of ingenuity, but as the saying goes, "art does not always pay". Particularly, where response and explanatory variables are involved, the response should match the explanatory set. In classification problems the correspondence of independent attributes to the class at least is implicit in the formulation. Farhangfar et al.[9] survey imputation methods in classification.

While relatively well studied,[1,4] MVs in data continue to perplex data mining practitioners, attracting monographs intended to bridge the gap, such as the texts by Enders,[2] Carpenter and Kenward.[19] Molenberghs and Kenward[20] afford a vast exposition of imputation methods in clinical trials.

Cheng et al.[21] use linear interpolation in gene expression analysis, which is a specialized area of research that cannot be placed into the categories of regression or classification. It is characterised by scarcity of data, and no clear distinction exists between instances and attributes. However, the data has some redundancy that can be exploited. A technique of bi-clustering in the instance-attribute space is used as the instrument to find localities where a higher precision of machine entered values can be achieved. Imputation and bi-clustering is performed one after the other in iterative manner until convergence. This processing mode resembles the one by Tseng.[15]

Most known MV imputation techniques treat numerical and categorical attributes differently, causing disarray when both types are present.[2]

3 Classifiers from nominal data

Classifiers from nominal / discrete data provide insight into how to deal with attribute value omission. Also, having obtained substitutes for MVs, a vehicle for result verification is required. In this section we adapt the Decision Branch (DB) and the Nearest Neighbour (NN) approaches for classification of nominal data and recall the Naive Bayesian (NB) method. A well-informed introduction to classification in a wider context of machine learning and data mining is found in the book by Kononenko and Kukar.[18]

3.1 Decision trees

The tree induction algorithms iD3, C4.5, C5 advanced by Quinlan[8] partition data in the form of a tree where branches are represented by values of a particular selected attribute. We will assume that all data is readily nominal/discrete. A complication arises when the attribute in question is continuous, requiring discretization at each node where the attribute is selected. Class value of a leaf node is statistically determined. An optimal tree is formed by selecting a feature that delivers the highest Information Gain (IG) for subsequent subdivision of data in a particular node. Partitions of high class mix are split earlier into separate branches in the course of tree induction. Here, we pursue a simpler version of the classifier where no model of data is ever learned but the leaf required by a test instance is directly evaluated each time by mining through the training set. This variety of decision tree is described next.

Algorithm 1: Decision Branch

Step 1. (*Initialization*). Set a minimum size for a sample to make inferences from.

Step 2. (*Direction*). Select a feature among features available for a particular test instance with the highest IG in describing data contained in the current leaf (the end node of the branch). Initial selection is carried out from all data excluding test instances, that is, the training set.

Step 3. (*Propagation*). Select the subset of data defined by the test instance best feature value. Check whether the number of instances in the new leaf is no less than the set minimum. Repeat recursively from Step 2 if it is not. Keep the shorter branch if it is.

Step 4. (*Generalization*). Evaluate the biggest class in the leaf and compare to the test instance. If the branch cannot be grown at all, rely on class prior probabilities at the root. Update classification accuracy for the test set. The accuracy for a particular class is the mean number of successes. Repeat from Step 2 until all test instances are streamed.

Details about training and testing sets, and techniques for classification accuracy calculation can be found in ref.[18] or other introductory texts.

Feature selection at the direction step of the algorithm is based on IG, a criterion, which measures mutual information between any two features. In the given context one of the attributes is always class. It is categorical; therefore, if the other feature is continuous, it is convenient to have it discretised. At the same time, IG relies on probabilities which can be estimated from frequencies, and so discretization is purposeful again. IG is applicable to all data, or any part of it, and measures how the class is influenced by a chosen feature, or vice versa - the higher IG, the stronger the influence. IG uses Entropy H - a quantity representing the average information contained in a feature. Three features participate in this calculation: the class, the attribute in question, and a joint feature of the two. With this in mind, Information Gain IG for class c and feature f is then simply:

$$IG(c, f) = H(c) + H(f) - H(c \times f),$$

where $\times$ denotes the Cartesian product of value sets. Entropy $H \geq 0$ for any feature f is obtained as follows:

$$H(f) = -\sum_{i=1}^{n} P(f = v_i) \cdot \log_2 P(f = v_i),$$

where P stands for probability, and $\log_2 P(f = v_i)$ is the morsel of information; $v_i, i = 1 \cdots n$ is a particular value of feature f present in a subset of data the formula applies to. Ref.[18] has more on the notions of Entropy and IG.

3.2 Nearest neighbourhood

The NN algorithm in this study is an adaptation for nominal data of the well-known k-NN technique.[22] It uses the Hamming loss for a distance function in the pseudo-space of data attributes.[18] The loss, which normally counts disagreement of attribute values between two instances being compared, is weighted by IG to make the space metric conform better to the data.[23,24] In the capacity of distance function the Hamming loss is also known as overlap metric.[24]

Weight setting in the distance formula is intended, generally, to reduce the influence of irrelevant attributes by making distances in their directions shorter. The pseudo-distance for nominal attribute space can then be expressed as follows:

$$d(p^1, p^2) = \sum_{i=1}^{n} g_i \cdot \delta(v_i^1, v_i^2);\ g_i = \frac{IG_i}{\sum_{i=1}^{n} IG_i};$$

$$\delta = 0, v_i^1 = v_i^2;\ \delta = 1, v_i^1 \neq v_i^2.$$

where p are data points in comparison; g are feature weights set by IG; v are feature values in instances p; $i = 1 \cdots n$ is feature index; and δ is the Kronecker's symbol to express incidence of two entities (although here 0 and 1change sides).

Algorithm 2: Nearest Neighbour

Step 1. (*Initialization*). Calculate IG for each feature to use as weights in the pseudo-distance formula. Adjust the weights to add to unity. Set parameter k - the number of closest instances drawn to establish class statistics for the neighbourhood of any test instance. Set a precision for distance calculations. The maximum distance for the pseudo-space is unity with the weighted Hamming loss. This makes using the same precision for different neighbourhoods throughout the space more consistent. By limiting the precision very similar instances are not neglected, which makes sampling more fair under the presumption of small sample size.

Step 2. (*Sampling*). Draw k closest instances to a current test instance (or having class assigned for the first time). Set radius of the neighbourhood by the furthest neighbour. Draw all instances in the neighbourhood within the radius according to the set precision.

Step 3. (*Classification*). Obtain the closest class to the test instance from mean distance statistics for classes with equally highest representation in the neighbourhood. Contest the class of the test instance and recalculate the accuracy. Repeat from Step 2 for each test instance.

One can see a similarity between the NN and DB classifiers and appreciate why weighting attributes by IG may work well. The result is largely determined by the top ranking attribute or having the highest weight in both approaches. The next attribute selected by DB may be further down the list than the second attribute due to redundancy. Away from the root of a tree, selection of attributes is driven by local considerations, whilst any particular neighbourhood keeps using globally assessed weights. The impact on results in either case is reduced due to reduced influence of subsequent choices; and both approaches have their weaknesses: NN approximates local weights with global weights, and DB loses certainty when successively selecting strongest features from the sample which gets smaller and smaller. However, while the NN concept is easily amenable to conversion to a tree structure, a pair of closest instances is not necessar-

ily found in the same leaf or even on the same branch.

3.3 Naive Bayesian

NB is a classical method for nominal data, widely used despite the requirement of attribute independence, given a class.[25,26] A presumption of this would be naive, generally. However, a simplification, which the conditional independence of attributes allows for, makes effectively more data available.

Given evidence x an optimal classifier should select class c_i with

$$P(c_i|x) > P(c_j|x), \forall j \neq i,$$

that is, the highest posterior probability. According to the classic formula attributed to Bayes

$$P(c|x) \cdot P(x) = P(x|c) \cdot P(c), \forall c.$$

Therefore, the alternative formulation of the minimum error rate decision rule is

$$P(x|c_i) \cdot P(c_i) > P(x|c_j) \cdot P(c_j), \forall j \neq i,$$

while $P(x) > 0$ cancels out. Because of the conditional independence

$$P(x|c) = P(x_1|c) \cdot P(x_2|c) \cdot \cdots \cdot P(x_n|c), \forall c,$$

where $x_k, k = 1 \cdots n$, are the attributes of x.

3.4 From real to nominal

A practical aspect associated with the application of classifiers from nominal / discrete data is discretization of real-valued attributes. A discretization method proposed by Yang and Webb[26] focuses on data abundance in any interval. The scheme is an interpretation of the conventional equal frequency method. The marginal probability density function in this method has a stepwise substitute. From this perspective, the predictions are more precise if more intervals subdivide the value range. These two contradicting aims have to be brought into a balance. Fixing the frequency at a certain level, according to ref.,[26] is the answer.

There is only an informal account in ref.[26] of how the Fixed Frequency discretization is to be performed. The method is evidently subdividing all attributes into the same number of intervals. The equal frequency is outwardly a simple concept, but handling of repeating values requires optimization to fulfil one of the prerequisites that the frequency across intervals has to be as even as possible, particularly to guarantee highest individual frequencies. Our realization of this "even", sooner than "equal", frequency concept is described next. We choose this method because it is easily implementable and has proven its utility in various applications.

Consider set A consisting of m points $a_j \in A \subset \mathbf{R}^n, j = 1 \cdots m$ $(m \geq 2)$. Having all attributes continuous in this representation, the algorithm discretises them in turn. The following steps apply to a current attribute $i \in 1 \cdots n$. It is assumed that each attribute values are sorted in an increasing order: $a_j^i \leq a_{j+1}^i, j = 1 \cdots m-1$. An important caveat to all this is that, while attribute values are real numbers, they can be mapped to integers due to limited precision data is generally known with, so no data point can be assumed unique.

Algorithm 3: Even Frequency Discretization

Step 1. (*Initialization*). Choose the target number of intervals, same for all attributes, $h_o > 1$. Initially each distinct value occupies its own interval. Set the number of intervals $h > h_o$ accordingly and calculate the mean frequency $f = m/h$.

Step 2. (*Repartition*). Find a pair of adjacent intervals so that their combined frequency is closest to f. If $h > h_o$ merge the intervals, and have $h \leftarrow h-1$ and f recalculated. Reiterate until $h = h_o$. The mean interval frequency can only increase in the repartition.

Step 3. (*Adjustment*). For each pair of connecting intervals adjust the boundary between them so that their frequencies were as close to each other as feasible without splitting repeating values. The mean interval frequency does not change in the adjustment. Reiterate until no further improvement is possible.

The repartition step addresses mainly the requirement of sufficiently high interval frequency and to a lesser extent the equality of frequencies between intervals. The adjustment step strives to resolve this inadequacy, and can also be applied over a random subdivision. However, advance repartitioning provides a good starting point.

4 Escaping missing values

It is sometimes possible to get around MVs by altering how the classification algorithm is implemented. This ability also holds clues to substituting MVs with surrogates. Transition to all categorical attributes assists the imposition of the required change, as it is often possible to apply by-coordinate moves in the algorithm. However, the general method is given by a mode of classifier implementation known as the "short memory", or the "lazy learning mode", whereby learning is preformed anew for each test instance. All previously described algorithms are implemented this way in the current work, which is clarified below.

The IG criterion is utilized in the DB and NN classification methods. The criterion involves two discrete attributes. This can be viewed as if instances having MVs in either of the two attributes were withdrawn for the purpose of a particular calculation. On the top of this, the DB method will simply have to choose features for which values are known when propagating a branch.

In the case of NN the situation is very different because

whole instances have to be matched, irrespective of absence of MVs. The notion of nearest neighbourhood is defined in terms of closest instances, and so one should be able to calculate distances between instances, even if not all attribute values are known. If this were possible, instances should be treated as less reliable, the more attribute values are missing, and applicable to a pair, the two instances should be regarded further apart. Calculation of distances in the pseudo-space of categorical attributes is based on match or no match of values. The probability of a mismatch in absence of some values when comparing two instance records in a given position has to be assumed high since the match is only one of all possible events. Making this a rule endows the distance calculation with the desired property.[9]

The "adaptation" of NB is that it facilitates the by-coordinate processing. The calculation is simply based on the probability by class for any particular attribute value. So, only class dependent subsets data defined by the value are involved. When classifying, attributes with no values in the test instance are simply not taken into account.

The discretization poses no problem at all since all attributes are individually treated. Any MVs are simply skipped.

5 Dummy surrogates

As we set to use nominal data only, MVs are easily entered as a special category. However, just by doing so, the problem is not solved but replaced with another one of dealing with noise. By using the special notation all noise is shifted to "missing" values. This guarantees correctly estimated probabilities for existing values if they are all genuine. Input from the dummy values is all-incorrect; however, it is sometimes possible to assume that their relative frequencies are, if misleading, then negligible, and not in favour of any particular class, that is, MVs are rare and random enough.

Uninformative ballast

Although it is a common practice to delete incomplete instances or attributes, this cannot be recommended as a general method, because only the training set can be dealt with in this way but not the test set. In addition, deletion of instances in the training set may distort the data distribution, and deletion of attributes may undermine legibility of the data, especially if presence of MVs is regarded admissible. However, the deletion of instances and attributes can be used to remove a redundant, systematic component in the MV pattern. Any remaining MVs can then be ignored. Because instances and attributes are both involved, a consideration should be given to which incomplete layer of the two to nominate for deletion and when. To minimize the impact a pre-processing step is proposed as follows.

Algorithm 4: Incomplete Information Dismissal

Step 1. (*Infogain*). Compute IG from available values for all features. Set a target for data reduction in terms of removed values, missing or not, relative to the size of original data (instances times attributes), which is a number, assumed small, between 0 and 1 .

Step 2. (*Layers*). Based on IG conveyed by individual values, calculate the amount of information contained in individual instances and attributes - layers of the data. A missing entry contributes no information.

Step 3. (*Prospecting*). Choose a layer containing the least information per value. To bridge between instances and attributes, select a matching sample comprising several instances or attributes, whichever number is currently bigger, satisfying the least information per value criterion.

Step 4. (*Dismissal*). Exclude the least informative layer from the data. Check the reduction rate. If the targeted minimum is achieved - quit, else - reiterate from Step 2.

A side effect of this algorithm is that it deletes attributes, whether incomplete or not. Indecisive overall, a feature may impart a substantial discriminatory power in a subset of data. So, complete attribute or not - same risks are taken. At the same time, the pre-processing doubles as a feature selection step.[3, 18]

6 Filling-in the blanks

A high redundancy of features in diagnostic domains suggests the "missing at random" scenario[1] whereby substitution of MVs from available data is always feasible. The substitution is possible if the classifier re-profiled for data entry has the ability to train and generalize without actually accessing MVs. It has to be able to classify any new instance in order to determine an appropriate subset of the training data, the substitutes can be evaluated from. Surely, the class is the main factor of subdivision. However, not all of the classifier training set must be labelled. This is reminiscent of modus operandi of Online Algorithms.[27, 28] Upon arrival of a new instance it is classified and added to a pool of critical instances, or the memory of the algorithm, where it is held unless subsequently it becomes clear it has to undergo a revision or be discarded as unreliable or redundant. The gist of the above is that class-unallocated instances are still valuable. This illuminates two distinct stages in the surrogate value entry. Firstly, MVs will have to be substituted in instances labelled for class. Secondly, the classifier will have to be trained on labelled instances and the class attribute set for not labelled instances by generalization. Any MVs in instances formerly without a class label can then be set from pertinent subsets of all data. Both stages will necessarily be iterative so changes could spread. The first stage applies to the labelled instances only and the second stage to all instances.

6.1 Contextual data completion by mode

Our algorithm performs completion and classification at the same time of any test instance by substituting attribute

modes for MVs from a sample of training data, which a classifier of choice queries to make the prediction. It is an iterative process of tuning instance records in turn to the remainder of the file that requires at least one valued instance per attribute per class. MVs are substituted from the subset having the same class as the test instance within the extracted sample. The first entry in a MV position is made regardless of whether the predicted class label is that of the test instance. The introduced value is only modified if subsequently the instance class becomes incorrectly predicted. If the test instance is not labelled, the embedded classification algorithm is used to assign the value. The iteration ends when the surrogate values stop changing or a limit for number of cycles is reached. The algorithm intends to produce the best possible array of substitute values within the classifier bias. However, substituted values in any given position on a record may differ for different classifiers. The absolute accuracy is implicit for the training set, yet the rule attaining this accuracy is unknown. Different classifiers approximate this rule differently. The selection of data a classifier makes has to be broad enough to include instances which are able to source a value for at least one MV. Less optimal classifier parameter settings are resorted to, when this is not possible. NB queries all data, and so the template for determining MVs is always in place. The sample consulted by classifier in the case of DB is a leaf and in the case of NN a neighbourhood where the test instance resides.

The flowcharts below outline how the data completion algorithm is applied when powered by different classifier engines (DB, NN or NB). The flowcharts are applicable to any of the two stages discussed above with little variation. Flow control constructs on these diagrams are as follows. "Initialize" refers to setting of parameters for the embedded learner and any preliminary actions. "First" and "Next" imply setting of the pointer for the stage dependent subset of incomplete instances. "Next" also asks whether all involved instances have been processed. "Current" symbolizes the beginning of the processing cycle for a particular incomplete instance. "InfoGain" shows where in the algorithm IG is evaluated for features available at that time. Block "Classify" applies to the subset where labels are unknown; otherwise, the classification is performed for testing only. "Select" refers to narrowing of the sample used for classification to instances of the current instance class. "Substitute" involves finding feature value modes in the selected data and filling the original blanks in the current instance record with the modes, where applicable. "Steady" checks whether the substituted values have all stopped changing; if yes, then "Quit" announces completion of a stage.

6.1.1 DB track

The outline of the technique involving DB appears in Figure 1. Using attribute modes for a particular class of data in a leaf causes instances of that class to bunch. If one of the substituted values defines a node before the leaf, the next time the branch is laid, the chance of correct identification of the test instance increases. Not only is the majority of instances transferred into a new leaf, other like instances are added. In short, substitution of MVs with applicable modes strengthens the affected attributes, promoting some closer to the root of the tree. Values of such an attribute then more diligently separate data into classes by virtue of IG. It is not clear from the chart, but the IG evaluation, classification and selection is recursively repeated in the sequence a number of times, causing the algorithm to slow down. As a trade-off, this provides the necessary "contraption" by which the selection can be easily adjusted to include particular class instances.

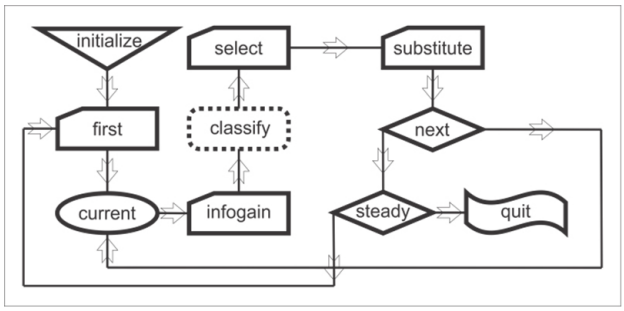

Figure 1: Flowchart of Decision Branch guided value submission

6.1.2 NN track

The principal scheme of the NN served algorithm is shown in Figure 2. Setting MVs by mode from data in the neighbourhood shifts the test instance towards instances the mode was obtained from, acting towards overall closer allegiance of the test instance with its class. The neighbourhood is stretched as required to include at least one instance of the test instance class. This algorithm evaluates IG once per cycle to metricise the space, which makes it much faster than the DB variety.

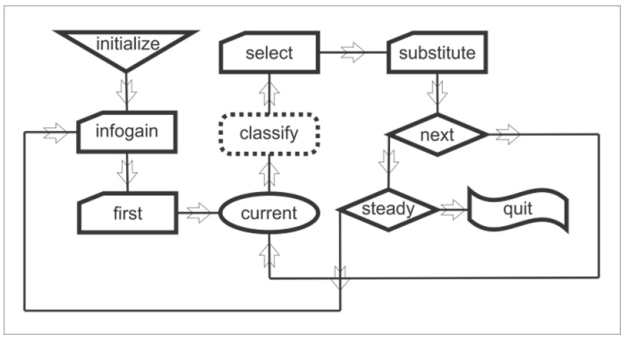

Figure 2: Flowchart of Nearest Neighbour assisted value submission

6.1.3 NB track

Flow of the NB based data completion method is very similar to that of the NN variety in Figure 2. The only difference is that no IG calculation is involved. This algorithm is the simplest of all, subject to the same constraints as for NB classification, and so is the fastest. Despite the strong assumption of class-wise independence of involved features, the theory behind NB is helpful in illustrating the validity of substituting MVs with pertinent modes. Suppose for a test instance x the only attribute with MV has index k. To guarantee selection of class c_i over any other by NB it has to hold that

$\frac{P(x_k|c_i)}{P(x_k|c_j)} \cdot \frac{P(c_i)}{P(c_j)} > 1, \forall j \neq i$

The best selection of x_k has then to

maximize $\frac{P(x_k|c_i)}{P(x_k|c_j)}$

The frequentistic approach to classification implies that for the attribute to be a perfect predictor its attribute values have to group by class. By this principle and in the view of attribute independence, selection of a mode for the winning class guarantees both, an increased probability in the numerator and a decreased probability in the denominator. Indeed, the selected value is then both: characteristic to class c_i and uncharacteristic to class $c_j, \forall j \neq i$; so it closely approximates the maximiser. To prevent the denominator from turning into zero, it has to hold that no attribute is a perfect predictor in respect of any value it takes; however, this is wider than it needs to be. Hence, by selecting class modes for all attributes with MVs, the instance class becomes reaffirmed.

6.1.4 Convergence

It can be easily seen that, if not for unknown class instances, a single iteration is sufficient for the NB based algorithm to converge when substituting MVs by class-wide attribute modes. Substituting by mode strengthens that mode. In this case there is a single cluster for each class. Different learners powering the principal scheme define different cluster systems. Within a cluster, instances reciprocate in cross-validation of their identity. As the mode spreads in the process, clusters consolidate. Therefore, the algorithm has a high propensity for convergence. In the case of NN, the use of distances has a tightening effect on a cluster, acting towards its isolation, and hence quick convergence to a unique solution. For DB, convergence is complicated by the dynamic nature of IG causing the whole tree to restructure and clusters within its leaves to change. Ideally, only a small number of strongest features define the structure, and only their IG change is of concern, but the sequence of their selection is unimportant. At the same time, using the strongest features selected by IG guarantees that classes do not dissipate in clusters. However, small leaf sizes, set parametrically, draw more structure defining features, their strength diminishing with propagation of a branch. In addition, IG evaluation, and so feature selection, becomes less reliable and modes unsubstantiated, leading in some cases to indefinitely continuing iteration switching back and forth between several possible solutions. Hence, there is a limit in place to make it stop. To conclude, an event of stochastic convergence is contributed by both, the data and the learner, particularly the learner parametric set. Notwithstanding, the algorithms applied in the current research did not iterate long, let alone reached the stipulated limit for number of cycles, when applied to examples discussed in the next section.

7 Simulation

In this section the proposed approach to data completion is applied to three different diagnostic problems, and results are evaluated through classification. Classification methods are those earlier introduced: DB, NN and NB. They are also components of the data completion principal scheme, giving rise to three different strains of the proposed method. Thus instantiated distinct methods and those used for benchmarking, described next, provide specifically for the categorical data representation, necessitating discretization of any continuous attributes by the Even Frequency method previously described.

7.1 Benchmarking

Performance of the proposed NB, DB and NN based completion methods is tested against seven benchmark methods. The first is to ignore MVs replaced with dummies, which may work well if MVs are rare or irregular, as in the state after instances or attributes abundant with MVs are removed. "Defaulting" MVs to most encountered, popular values of their attributes is the second benchmark. The class attribute can be dealt with in the same manner. The third benchmark is classification skipping MVs. The methods are denoted "Ignore", "Default" and "Skip", respectively, in the depiction of results (Figures 3 & 4).

The next three methods illustrate the approach pursued in ref.[7] for decision trees and in ref.[9] for the NB classifier. Here, we expand it to include all aforementioned classifiers, although in the more narrow context of all-nominal / discrete data. The omnibus routine visits each attribute in turn, evaluating any MVs by engaging a particular classifier (NB, DB or NN) while using all other attributes (including the actual class) as predictors. This is possible because all three classifiers have the ability to generalize in presence of MVs in the training set. It is essentially the same approach as in the third benchmark method, except that MVs are also substituted with values predicted from the data with original non-entries. Any MVs in the actual class attribute are dealt with exactly in the same manner. The three methods are tagged "Omnibus" NB, DB or NN, respectively, in the graphical representations of results (Figures 3 & 4).

The seventh comparison method descends from the NN variety of the proposed completion method, which we extend to include the class attribute. This kind of handling goes by the name of "Hot Deck".[2] With this approach the search becomes truly unsupervised and is performed in the extended space of attributes where any MV is substituted purely from considerations of pattern similarity, regardless of the class. However, the space metric is the same as in the NN method. It is normally assumed that class domains are separable in the feature space, and in the extended space they are even more so. A flowchart representation of this algorithm would be the same as in Figure 2, except no classification is performed, and so the selection extends to all sampled instances. The "Hot Deck" variety is titled accordingly in the results (Figures 3 & 4), while the three alternative designs of our principal method of Contextual Data Completion by Mode are marked "Context" NB, DB or NN, respectively.

7.2 Data

Datasets used for the current evaluation are characterized in Table 1. The DiScRi (Diabetes Screening Research Initiative) data is a proprietary one, from research conducted at Charles Sturt University, made available to the Federation University Centre for Informatics and Applied Optimization Health Informatics Laboratory. It represents a collection of medical records containing information associated with diabetes and its complications. Other examples are sourced from the UCI Machine Learning Repository,[29] publically accessible via the Internet. The Horse Colic dataset offers a veterinary example, to train a classifier on various livestock health indicators and to determine whether a particular lesion is surgical. The data is from McLeish and Cecile, then of University of Guelph, Ontario, Canada. The Secom dataset presents an industrial diagnostics example. It features output from a system monitoring a semi-conductor manufacturing process complete with a check on the items, whether they are functional, and was made available by McCann and Johnson. No dataset includes attributes that are either unrelated to classification problems featured within, or are single-valued, as shown. All examples in Table 1 comply with the requirement that at least one value per attribute per class has to be known.

Table 1: Dataset dimensions

Datasets	Attributes			Instances		Values
	All	Numerical	Incomplete (%)	All	Incomplete (%)	Missing (%)
DiScRi	97	54	64	824	100	31
Horse Colic	21	7	95	368	98	28
Secom	475	474	89	1567	100	6

Apart from Diabetes Mellitus (DM), the DiScRi dataset can also be used to predict Cardiovascular Disease (CVD) or Hypertension (HT). Attributes of the DiScRi dataset as is, without featuring use of medication, however, give best support to DM, less support to HT, and only some support to CVD. Attending to all problems simultaneously limits candidate values a particular MV may espouse, as the one nominated has to be same for all problems. On this occasion, our evaluation effort exploits the connection between DM and HT but does not extend to CVD. A focused study of the diagnostic problem, including all three components, was conducted by us earlier,[30] where the data is discussed at length.

7.3 Diagnostic domains

Class structure of the datasets is shown in Table 2. All datasets have the control class, numbered 0, and the diagnostic class, numbered 1. It is common to call the classes also normal / negative and abnormal / positive, respectively, although a clarification is required depending on the context. In the Horse Colic case what is regarded normal and abnormal is, for instance, counterintuitive. Generally, however, a number of abnormal conditions may be targeted. For instance, several pre-diabetic conditions and three subtypes pertain to diabetes (gestational, type 1 and type 2 diabetes mellitus). The chosen example focuses on Type 2 DM, so it is Class 1; anything else is in Class 0. It is not unusual for diagnostic domains that Class 0 is large, which is evident from Table 2, although where the class is only partially known, the dominance of Class 0 is not obvious. This exacerbates the difficulty of predicting abnormal instances, because classifiers get insufficiently trained for the class of diagnostic interest. For example, the NB classifier relies substantially on prior probabilities in decision making. Another effect associated with diagnostic problems is that the normal class - using the metaphor of attribute space - tends to surround the abnormal. The diagnostic class concentrates ideally about a single point, whereas the control, comprising non-diagnostic instances of all sorts, has multiple congregation points. This would represent a difficulty for the NN classifier. For that reason diagnostic problems are also known as One Class problems. Generally, correctly alerting to an abnormal situation is more valuable than the opposite. Thus, the size disparity and the "overhanging" effect increase the costs of inaccurate predictions.

Table 2: Data subdivision by class

Problem	Classes (%)		
	Unknown	0	1
Diabetes Mellitus Type 2	0	74	26
Hypertension	20	32	48
Horse Colic	0	63	37
Secom	0	93	7

Kubat and Matwin[31] address the problem by one-sided selection of the positive instances, that is, by disregarding the negative instances close to the class boundary. This is a proper approach as the actual instance numbers should

not bear as heavily on classification, but somewhat different problems are encountered from time to time, and therefore solutions continue to be actively sought. Yoon and Kwek[32] go further by proposing under-sampling of the majority class throughout the dataset. Nguyen et al.[33] exercise a more flexible approach, whereby the positive and negative class sets are provisionally clustered before a classifier is constructed, effectively balancing the data. Gonzalez-Reyna et al.[34] in their traffic sign recognition problem apply techniques that, instead of under-sampling of the majority class, over-sample the minority class or do both, particularly the technique by Chawla et al.[35] We proposed solving of the imbalanced data problem by class noise reduction in ref.[10]

Despite being very class imbalanced, the Secom example has many instances and attributes, posing a challenge for quick evaluation. Therefore, only half of the dataset in Table 1 was used. The representative sample was obtained by drawing instances at random proportionally to class sizes. This generic technique of data reduction approximately halved the number of all MVs at the same time. We used another technique of this sort when condensing the temporal information associated with patient visits - a precursor of the DiScRi data in Table 1.[30] About a quarter of the original MV content was thus reclaimed.

7.4 Evaluation and comparison

Performance of the proposed data completion methods is judged against benchmarks by diagnostic accuracy of conditions within data. The results for the DiScRi DM and HT problems are illustrated in Figure 3. Likewise, the results for Horse Colic and Secom examples appear in Figure 4. The charts are based on simple mean accuracy for Classes 0 and 1, that is, half-sum of the proportions of successes for test instances of each class, known as balanced accuracy. Kubat and Matwin[31] advocate using geometric mean of class accuracies for the same purpose. Using means of either type allows for a grand view of performance. For a better understanding, though, predictability of Class 0 is usually higher than that of Class 1, in accordance with what was earlier noted. If data-set or feature-set varies, or a different classifier is applied to the imbalanced data, small loss in accuracy for Class 0 usually means big gain for Class 1, and vice-versa. A connotation of the balanced accuracy is that the single indicator is sufficient and it would reflect only significant changes under the circumstances. The accuracy is estimated via the leave-one-out cross-validation resampling,[18] a technique commonly used when a comparison between methods is involved. In all mentioned charts the accuracy is plotted three times for each classification method: once for full data (not to be confused with having all MV positions filled) and twice for trimmed data. Reduced datasets result from Incomplete Information Dismissal – the proposed preprocessing. Fractions denoted as "half" and "third" refer to targeted data reduction rates of at least 50% and 70%, respectively.

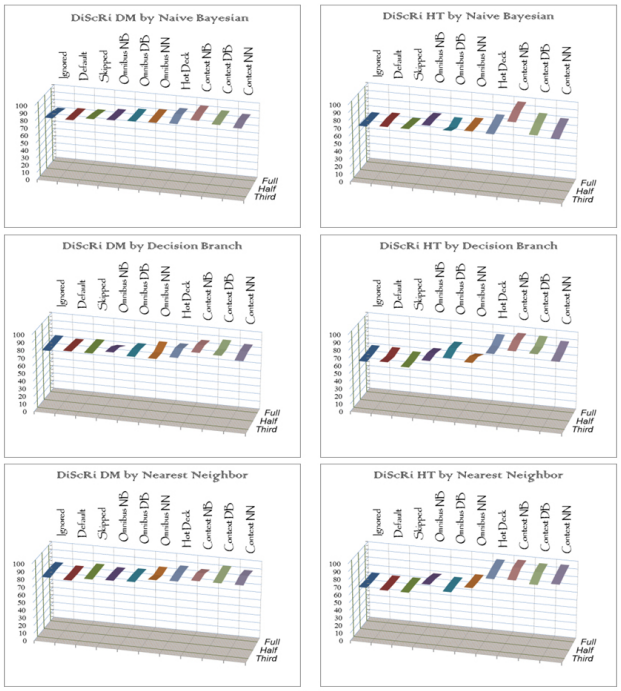

Figure 3: Balanced accuracy applicable to DiScRi DM (left) and HT (right) problems for different classifiers on full data and two fractions of data content after completing data using the array of methods

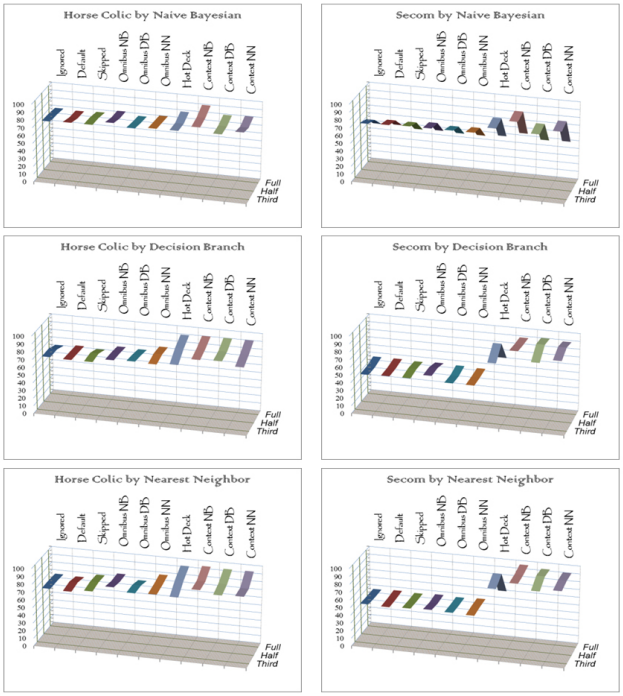

Figure 4: Balanced accuracy applicable to Horse Colic (left) and Secom (right) problems for different classifiers on full data and two fractions of data content after completing data using the array of methods

7.4.1 Full data

We present discussion of full dataset results first. It can be observed that outcomes of the first two benchmark methods of dealing with MVs are substandard in all presented problems. All examples clearly exhibit the attribute pattern of inundation of data with MVs, rather than the instance one - something inherent in diagnostic problems. Hence, we could say that attributes abundant with MVs become predominantly irrelevant. In both cases classification relies on attributes originally more filled with data and the results of ignoring or defaulting are not much different. When, instead, MVs are skipped, this has a bearing on precise estimation of involved probabilities. However, the classifiers rely on data that is contextually available, and so the attributes inundated with MVs are largely left out. Unsurprisingly, the results are only slightly better than for the two preceding methods.

Omnibus NB, DB or NN data completion in all examples offers only some improvement. It is difficult to comment on the current results because they depend strongly on the numbering of instances and attributes in data since any strong association between attributes should be assumed circumstantial. Therefore, the margins are often insufficient to confidently substitute a MV with a single label, rendering the output semi-random. Perhaps for this reason Farhangfar et al.[9] attempt to boost the result by filtering out infrequent substitutes. Evidence of this indeed occurring is that no particular method among the three performs consistently better than others. It can also be observed that when data is reduced the second time the accuracy does not necessarily degrade. Although, the last result may be due to stronger, unadulterated ties between features and the class, existing after many spurious influences were eliminated.

Predictability of either normal or abnormal outcomes in all examples increases dramatically after completing the data in accordance with the proposed method and regardless of the engine used (DB, NN or NB). However, the Hot Deck variety offers a "no-frills" alternative to any of the three brands of the formal approach, adjusting for its reliability and speed.

Performance of the NB based data completion method stands out. The algorithm is not only faster but often more accurate than the other two, DB or NN powered, which is at odds with the basic nature of NB. For medical data this can be linked to its structure: highly relevant attributes are inundated with MVs, while unimportant ones are largely not. The situation is likely due to the high costs associated with specialized testing. The breadth of mode selection in the algorithm causes high contraction of data distribution in the pseudo-space of attributes, that is, classes become identified by the distinct feature modes. Thus, the data becomes more compliant with the assumption of NB that attributes have to be class-wise independent. When a problem is split, as is the case for DiScRi DM and HT, MVs have to be identically substituted in any part, and the surrogates arising from NB may appear to be more centred than by other methods, so no sub-problem is disadvantaged more than the other.

Across the board, the DB guided MV submission hardly registers an advantage over the NB based one, although room for improvement may be restricted due to data bias and the predictability that is already high. Decision trees are generally praised more for comprehensibility of learned rules than for their accuracy. On the other hand, the DB directed completion method is appealing for its design rendering selection of substitutes more realistic. NN, as implemented, is notionally similar to DB, so one should expect a similar performance when the two are components in turn of the proposed data completion scheme. However, the NN assisted completion method, as observed, is notably behind its counterpart. This may be so because NN adheres to the data distribution closer than NB or even DB. So, instead of neutralizing any negative impact of an entry in a MV position, which is the primary objective, the completion method harnessing NN sooner reconstructs data.

When comparing different completion methods one has to make sure to discount evidence from a classifier that is also the engine of a particular completion method. A qualification is also required when not all data instances are class-labelled. In particular, these instances are not taken into account when the Ignored and Skipped MVs benchmark methods are applied. Among the test problems we attempt, DiScRi HT is the only example of this kind. All proposed in the current work data completion methods show a much higher accuracy on the DiScRi HT than on the DM problem, something to expect under the circumstances. For a better overall view of performance of the data completion algorithms on full datasets, it is also helpful to watch indicators for reduced datasets, in the sense that they provide an extended sample of results.

7.4.2 Trimmed data

We next analyse the results obtained for reduced datasets. It can be observed that the attribute pattern of MV inundation is prevalent in all examples, and perhaps this is common in diagnostic problems. When targeting 50% or 70% reduction we obtain all instances intact except for the number of attributes. Withheld attributes are either less informative or have many MVs. Either way, trimming by 50% appears to be safe for all featured problems, with results even better in some cases compared to the full data. For the DiScRi problems the results are almost unchanged. Some drop of accuracy is observable in the Horse Colic example with any method of data completion. This example has fewer features by an order of ten compared to the other two datasets, and so reduction of the feature-set has a much stronger impact. In the Secom example there is a conspicuous change when the data is classified with NB. Both classes get evenly predicted.

ISSN 1927-6974 E-ISSN 1927-6982

What was compounded by the imbalanced class situation becomes resolved in the reduced attribute setting, which is characteristic for NB since the dependence between attributes is reduced as well. Other classification results for the reduced Secom data are almost the same as for the full data.

Reduction yield in terms of MV numbers ratio in reduced and full data is shown in Table 3. The yield depends strongly on the MV pattern in the data, and therefore does not necessarily resemble the targeted data reduction rate by 50% or 70%. However, the yield is impressive in all examples, allowing for taking a large number of MVs out of consideration. Compared to trimming by 50%, trimming by 70% makes the results almost invariably negatively budge, although not by much. Nevertheless, the trade-off for the lost accuracy is worthwhile.

Table 3: MV reduction yield (%)

Data	Full	Half	Third
DiScRi	0	34	68
Horse Colic	0	58	82
Secom	0	63	80

7.4.3 Discussion

From the results charted in Figures 3 & 4 it is evident that diagnostic predictions are better after data completion under the proposed approach. While each classification method has its own merits and demerits, our observation is that, apart from better estimation of involved probabilities, data completion effectively extends the feature-set. In Figure 5, NN accuracy on full data is compared before and after completion by the DB powered method.

The NN classifier outperforms others. Despite this, choosing NN for obtaining the overall picture of improvement, exemplified by Figure 5, is of secondary importance. This does not magnify the effect and ensures that the incomplete data is not misrepresented. Indeed, only the difference in results before and after completion is of interest. Of primary importance is the completion method, though, and the DB based method does not appear to be the best. So, the picture can be described as overall typical. Employing NN for classification in this connection has another justification: the classifier has to be different from the one used in the completion method for the fairness of representation. On the other hand, use of NB as the classifier or the engine of completion method is theoretically limited.

However, the proposed method is consistently better with any engine when comparing the state after completion to the state before completion. To assert this requires a picture for the worst case scenario. Of what we know, data completion with NN is more conservative than by DB or NB, and the DB classifier is no better than NB. Note that calculation of probabilities in NB is more elaborate than in DB. The last consideration also applies to the NN classifier where use of distances enhances estimation of probabilities, purely frequency based in DB. To sum this up, exactly the opposite satisfies the specification of an unfavourable outcome. The corresponding accuracy chart is shown in Figure 6 where data completed to NN is classified by DB. Clearly, the improvement is about the same as in Figure 5, if not the accuracy.

Figure 5 draws on results for full data as plotted in the last rows of Figures 3 & 4. Likewise, Figure 6 interprets results in the middle rows of Figures 3 & 4. The darker, bottom portion of each element in the two figures corresponds to the Skipped MVs treatment when the classification algorithm of choice is applied. The lighter, upper portion of any bar is the projected gain of accuracy by the same classifier if missing data is filled to the corresponding variety of the proposed scheme. The middle section of the Secom element represents a correction, as further explained. In reading these figures one should realise that the effective origin is located not at 0% but at 50% accuracy rate due to the likely data imbalance, reaching the level of stark contrast when Class 0 is fully authenticated and Class 1 not at all.

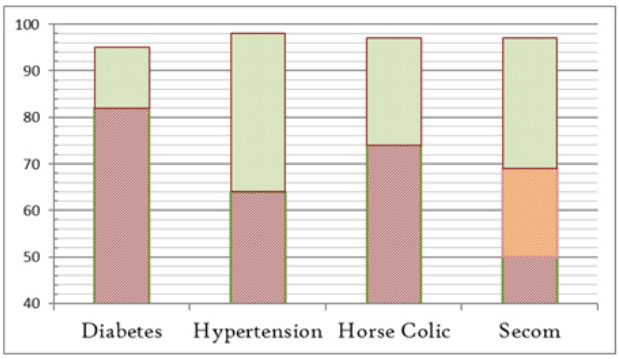

Figure 5: Improved predictability by NN of any outcome, normal or abnormal, after DB guided MV submission in full data

Data with many MVs is too flexible, so a high resulting accuracy may be unwarranted. Although, a misrepresentation may occur: the accuracy gain in the Secom case, as evident from Figures 5 & 6, is actually not that large, if one observes the Skipped MVs accuracy by NB in Figure 4 (first row, last column). It improves after trimming, which explains nature of the middle section of the chart element in Figures 5 & 6. Generally, with more classes specificity of MV substitute values should increase. However, there are only two classes in examples discussed in the current research. Some solace can be found in the small size of the diagnostic class. Apart from the classification bias, there is nothing else in the problem to focus the selection more. However, the circumstance that the same data provides for different problems can be exploited to greatly limit applicable surrogate ranges. Substituted MVs in linked constituent problems are expected

to be identical. To make this applicable to the DiScRi DM and HT problems, we staged a super problem where classes are defined by Cartesian product of class value sets of the components. Each combination of class values in the underlying problems corresponds to a class in the super problem, thereby restricting the choices that are available for surrogate entry. Cartesian attribute products in classification were made popular by Pazzani.[36]

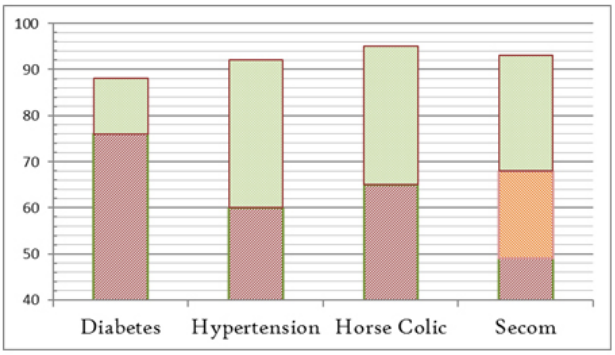

Figure 6: Improved predictability by DB of any outcome, normal or abnormal, after NN assisted MV submission in full data

It is plausible that multiple attributes exacerbate the imbalanced situation pertaining to diagnostic data. This is due to the "curse of dimensionality",[18] the phenomenon owing to which data space becomes less populated, the more dimensions it has. The minority class suffers more in this transition. Conversely, the opposite should improve the situation. It is peculiar that the NB classifier was able to sense the change and no other classifier could in the Secom example. It came to the fore with any method of MV handling, including the benchmarks, and kept getting better after more trimming. Surely, with many features removed the theoretical applicability of NB has improved. Nonetheless, the class numbers and associated prior probabilities in the NB formula did not change. Unlike in other examples the Secom data imbalance is very large. This requires a better explanation and cannot be made from a single observation. However, NB is expected to cope well with noise for the wide net it casts, and this is a promising lead. In all other examples the imbalanced situation is mild. It clearly gets improved after data completion using the proposed methods, with no exception as to data, and the Secom example, despite having the least rate of missing values, makes this especially clear. This has to be credited to the main property of the proposed methods whereby data becomes less noisy through better accommodation of instances in their class domains. The approach is completely different, though, from the one pursued by us in ref.[10]

8 Conclusion

In this work we applied a number of techniques of missing value imputation to some typical datasets from the domain of diagnostics. The proposed methodology can be used on data of any type via discretization of continuous attributes. However, discretization is instrumental for getting around missing values when classifying data, and so for data completion. As a consequence, the fitted values acquire the desired subtlety. Indeed, no missing value can generally be made exactly known. Often substitute values have little or no impact on classification. The attributes containing them may be of little relevance, or there are salient, well-defined features predetermining the outcome. Likewise, a potentially strong feature conveys little information when inundated with missing values. A proposed pre-processing step, which removes weak features, saves a lot of effort in dealing with missing values. The purpose of missing value submission in this work is seen not as guessing of correct values but removal of hindrance that incomplete data creates for classification. The completion, as performed, intends to reveal data at its best within ability of a learner in its core. The amount of missing values is sometimes huge, so the data conformation may be far fetching. There seems to be a "spectre" haunting the domain of diagnostics. Missing value input calls for better framing. In this work a convoluted representation was exercised in one example where several known diagnostic problems arising from the same data were joined together by means of Cartesian product to arrive at substitute values shared by all.

Acknowledgements

The authors thank all referees who supplied comments helping to improve this paper.

References

[1] Little R.J.A., Rubin D.B. 2002. Statistical analysis with missing data. 2nd ed. Wiley.

[2] Enders C.K. 2010. Applied missing data analysis. Guilford.

[3] Bagirov A., Yatsko A., Stranieri A., Jelinek H.F. Feature selection using misclassification counts. In Proceedings of the 9-th Australasian Data Mining Conference (AusDM 2011). 2011; 121: 51-62. Conferences in Research and Practice in Information Technology.

[4] Schafer J.L. 1997. Analysis of incomplete multivariate data. Chapman and Hall.

[5] Latkowski R., Mikolajczyk M. Data decomposition and decision rule joining for classification of data with missing values. In Transactions on Rough Sets, Lecture Notes in Computer Science, LNCS 3100. 2004: 299-320. Springer.

[6] Sterne J.A.C., White I.R., Carlin J.B., Spratt M., Royston P., Kenward M.G., et al. 2005. Knowledge discovery from legal databases. Springer.

[7] Rahman G., Islam Z. A decision tree-based missing value imputation technique for data pre-processing. In Proceedings of the 9-th Australasian Data Mining Conference (AusDM 2011). 2011; 121: 41-50. Conferences in Research and Practice in Information Technology.

[8] Quinlan R. 1993. C4.5: Programs for machine learning. Morgan Kaufmann.

[9] Farhangfar A., Kurgan L., Dy J. Impact of imputation of missing values on classification error for discrete data. In Pattern Recognition. 2008; 41: 3692-3705. http://dx.doi.org/10.1016/j.patcog.2008.05.019

[10] Stranieri A., Yatsko A., Golden I., Mammadov M., Bagirov A. Capped k-NN editing in definition lacking environments. In Journal of Pattern Recognition Research. 2013; 8(1): 39-58. http://dx.doi.org/10.13176/11.465

[11] Gamberger D., Lavrač N. Filtering noisy instances and outliers. In Liu H., Motoda H. (editors) Instance selection and construction for data mining. 2001: 375-394. Kluwer.

[12] Junninen H., Niska H., Tuppurainen K., Ruuskanen J., Kolehmainen M. Methods for imputation of missing values in air quality data sets. In Atmospheric Environment. 2004; 38: 2895-2807. Elsevier. http://dx.doi.org/10.1016/j.atmosenv.2004.02.026

[13] Wang Q., Rao J.N.K. Empirical likelihood-based inference in linear models with missing data. In Journal of Statistics. 2002; 29: 563-576.

[14] Zhang S., Zhang J., Zhu X., Qin Y., Zhang C. Missing value imputation based on data clustering. In Transactions on Computational Science. 2008; 1: 128-138. Springer, LNCS 4750.

[15] Tseng S.M., Wang K.H., Lee C.I. A preprocessing method to deal with missing values by integrating clustering and regression techniques. In Applied Artificial Intelligence. 2003; 17: 535-544. http://dx.doi.org/10.1080/713827170

[16] Peng C.Y.J., Harwell M., Liou S.M. Ehman L.H. Advances in missing data methods and implications for educational research. In Sawilowsky S.S. (editor) Real data analysis. 2007: 31-78. Charlotte, NC: IAP.

[17] Xu S., Schroedera E.B., Shetterlya S., Goodricha G.K., O'Connor P.J., Steinera J.F., et al. Accuracy of hemoglobin A1c imputation using fasting plasma glucose in diabetes research using electronic health records data. In Statistics, Optimization and Information Computing. 2014; 2: 93-104.

[18] Kononenko I., Kukar M. 2007. Machine learning and data mining: introduction to principles and algorithms. Horwood.

[19] Carpenter J., Kenward M. 2013. Multiple imputation and its application. Wiley.

[20] Molenberghs G., Kenward M. 2007. Missing data in clinical studies. Wiley.

[21] Cheng K.O., Law N.F., Siu W.C. Use of biclustering for missing value imputation in gene expression data. In Artificial Intelligence Research. 2013; 2(2): 96-108.

[22] Keller J.M., Gray M.R., Givens J.A. A fuzzy k-nearest neighbour algorithm. In IEEE Transactions on Systems, Man and Cybernetics. 1985; 15(4): 580-585. Reprinted in (1991) Dasarathy B.V. (editor) Nearest Neighbor (NN) norms: NN pattern classification techniques; IEEE.

[23] Daelemans W., Van-Den-Bosch A. Generalization performance of backpropagation learning on a syllabification task. In Proceedings of TWLT3 - the Third TWENTE Workshop on Language Technology. 1992: 27-37. Enschede, Netherlands. Morgan Kaufmann.

[24] Wettschereck D., Aha D. W., Mohri T. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. In Artificial Intelligence Review. 1997; 11: 273-314. http://dx.doi.org/10.1023/A:1006593614256

[25] Domingos B., Pazzani M. Beyond independence: conditions for the optimality of the simple Bayesian classifier. In Proceedings of the Thirteenth International Conference on Machine Learning. 1996: 105-112. Morgan Kaufmann.

[26] Yang Y., Webb G.I. Discretization for naive-Bayes learning: managing discretization bias and variance. In Machine Learning. 2009; 74(1): 39–74. http://dx.doi.org/10.1007/s10994-008-5083-5

[27] Aha D.W., Kibler D.W., Albert M.K. Instance-based learning algorithms. In Machine Learning. 1991; 6: 37-66. http://dx.doi.org/10.1007/BF00153759

[28] Wilson D.R., Martinez T.R. Reduction techniques for instance-based learning algorithms. In Machine Learning. 2000; 38: 275-286.

[29] UCI Machine Learning Repository. Available from: http://mlearn.ics.uci.edu/. University of California, Irwin, USA

[30] Jelinek H.F., Yatsko A., Stranieri A., Venkatraman S. Novel data mining techniques for incomplete clinical data in diabetes management. In British Journal of Applied Science and Technology. 2014; 4(33): 4591-4606. http://dx.doi.org/10.9734/BJAST/2014/11744

[31] Kubat M., Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. In Proceedings of the Fourteenth International Conference on Machine Learning. 1997: 179-186. Morgan Kaufmann.

[32] Yoon K., Kwek S. A data reduction approach for resolving the imbalanced data issue in functional genomics. In Neural Computation and Applications. 2007; 16: 295-306. http://dx.doi.org/10.1007/s00521-007-0089-7

[33] Nguyen G.H., Bouzerdoum A., Phung S.L. A supervised learning approach for imbalanced data sets. In Proceedings of ICPR 2008 - the Nineteenth International Conference on Pattern Recognition. 2008; 4: 1-4. IEEE.

[34] Gonzalez-Reyna S.E., Martinez-Trinidad J.F., Carrasco-Ochoa J.A., Avina-Cervantes J.G., Ledesma-Orozco S. Applying balancing techniques in traffic sign recognition. In Artificial Intelligence Research. 2014; 3(4): 38-42.

[35] Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002; 16: 321-357.

[36] Pazzani M.J. Constructive induction of Cartesian product attributes. In Liu H., Motoda H. (editors) Feature Extraction, Construction and Selection: A Data Mining Perspective. 1998: 341-354. Kluwer.