Artificial Intelligence in Medicine 55 (2012) 37–50

Contents lists available at SciVerse ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

A classifier ensemble approach for the missing feature problem

Loris Nanni[a,*], Alessandra Lumini[b], Sheryl Brahnam[c]

[a] Department of Information Engineering, University of Padua, Via Gradenigo, 6/B, 35131 Padova, Italy
[b] DEIS, University of Bologna, Via Venezia 52, 47521 Cesena, Italy
[c] Computer Information Systems, Missouri State University, 901 S. National, Springfield, MO 65804, USA

ARTICLE INFO

Article history:
Received 2 February 2011
Received in revised form
25 November 2011
Accepted 26 November 2011

Keywords:
Missing values
Imputation methods
Support vector machine
Fuzzy clustering
Data corruption
Equipment malfunctions

ABSTRACT

Objectives: Many classification problems must deal with data that contains missing values. In such cases data imputation is critical. This paper evaluates the performance of several statistical and machine learning imputation methods, including our novel multiple imputation ensemble approach, using different datasets.
Materials and methods: Several state-of-the-art approaches are compared using different datasets. Some state-of-the-art classifiers (including support vector machines and input decimated ensembles) are tested with several imputation methods. The novel approach proposed in this work is a multiple imputation method based on random subspace, where each missing value is calculated considering a different cluster of the data. We have used a fuzzy clustering approach for the clustering algorithm.
Results: Our experiments have shown that the proposed multiple imputation approach based on clustering and a random subspace classifier outperforms several other state-of-the-art approaches. Using the Wilcoxon signed-rank test (reject the null hypothesis, level of significance 0.05) we have shown that the proposed best approach is outperformed by the classifier trained using the original data (i.e., without missing values) only when >20% of the data are missed. Moreover, we have shown that coupling an imputation method with our cluster based imputation we outperform the base method (level of significance ~0.05).
Conclusion: Starting from the assumptions that the feature set must be partially redundant and that the redundancy is distributed randomly over the feature set, we have proposed a method that works quite well even when a large percentage of the features is missing (≥30%). Our best approach is available (MATLAB code) at bias.csr.unibo.it/nanni/MI.rar.

1. Introduction

In practice, data are often missing, cases incomplete. People are unable or unwilling to answer all survey questions, pixels fail, sensors are bad, equipment malfunctions, certain medical tests cannot be performed for a variety of reasons. These and many more circumstances, such as unusual readings, extreme noise, and data corruption, can introduce values that can greatly bias the estimation and prediction process of decision support systems that rely on advanced machine learning methods, such as artificial neural networks. The relationship between these methods and the data is complicated.

Systems performance is strictly related to rates of missing data: a rate less than 5% are generally considered manageable, while rate of 5–15% requires ad hoc methods, and rate larger than 15% may be very hard to handle. To evaluate the complexity of the data deficiency in a missing value problem, some statistical models have been proposed in the literature [1] with the aim of measuring the randomness of the missing. The most used models to measure problem complexity are: missing completely at random (MCAR), i.e., the missing probability for a random variable X is independent of the actual value of X or the values of the other features; missed at random (MAR), i.e., the missing probability is independent of the value of X after controlling the other variables [1]; not missing at random (NMAR), i.e., the missing probability for a random variable X could depend on the value of that variable. Even if MCAR is preferable, in many real-world applications, MAR is a more realistic model [1].

In some cases, algorithms cannot work with missing data. In other cases, missing values can result in wrong decisions, a drawback that is particularly critical in medical applications, where, for instance, a wrong treatment can lead to the death of a patient. Unfortunately, the most common strategy for dealing with absent values in these systems is essentially to ignore them [2]. This technique, commonly referred to as filtering, clearly becomes infeasible in case of a high percentage of missing. For example, in the

* Corresponding author.
E-mail addresses: loris.nanni@unipd.it (L. Nanni), alessandra.lumini@unibo.it (A. Lumini), sbrahnam@missouristate.edu (S. Brahnam).

0933-3657/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2011.11.006

field of biological data, epistatic miniarray profiling (E-MAP) [3] is a method for analyzing comprehensive genetic-interaction maps usually represented in the form of a symmetric matrix but often with a large number of missing values (even >30%). In such a case simply discard the genes that contain missing values in not feasible, since more than 90% data in the E-MAP matrix could be removed.

A more feasible solution is based on the so-called *imputation*, which is the substitution of a missing value with a meaningful estimate. Typical methods for data imputation are based on replacing the missing value with the most similar among existing data points (i.e., *hot-deck* imputation [4]), or the mean of that feature across all the training data or limited to the k-nearest neighbors.

Several robust statistical methods have been developed to address the problem of missing values: multiple imputation (MI) is one of the most common methods for handling missing values [5]. MI is a technique which replaces the missing values by their $m > 1$ simulated versions (m is typically small). Each of these completed simulated datasets is analyzed using standard methods. The results are then combined to produce estimates and confidence intervals that incorporate the uncertainty introduced by the missing data.

The missing data imputation is an area of statistics that has attracted much attention in recent decades, for survey on the most important works see [6–8].

Recently machine learning imputation methods have been developed. They estimate missing values by constructing a predictive model to estimate the absent values from information in the dataset. Some well-known stand-alone learning algorithms that have been applied to this problem include the multi-layer perceptron (MLP), k-nearest neighbors (KNN), self-organizing maps (SOM) and the decision tree (DT). Other approaches are based on Bayesian networks [9,10]. For example, in [10] two Bayesian methods for imputation are proposed, based on the construction of a Bayesian network for each attribute with missing values.

In [11] a review and comparison, based on real datasets, of existing methods (including single imputation, likelihood-based multiple imputation, probabilistic split and surrogate split) for coping with missing data in decision trees is reported, which found multiple imputation to be the best of the existing methods investigated. In [12] an iterative boosting method for improving the quality of the imputed features is proposed and in [13] the same authors studied the influence of the imputation of missing features on the classification error for five methods, showing that, in general, imputation is beneficial for the classification of objects with missing features. In [1] the authors have compared some imputation methods (including commercial methods for multiple imputation as Amelia II, WinMICE and SAS), reporting that only the results obtained using machine learning-based techniques (i.e., MLP, KNN and SOM) were significantly better than those in which records containing missing values are eliminated. Moreover, they report that all the three machine learning-based techniques have a very similar performance.

A very interesting machine learning approach is proposed in [7], where they study the random subspace approach for the missing feature problem. Their approach, which they name Learn++.MF, it based on the distribution update concepts of Learn++ with the random feature selection of random subspace (RS). When an instance with missing features needs to be classified, only those classifiers trained with the features that are presently available in that test pattern are used for the classification.

Learn++.MF makes two assumptions:

- The feature set is partially redundant.
- The redundancy is distributed randomly over the feature set.

In [7] the authors compare their Learn++.MF with the one-class approach [14], with the expectation–maximization (EM) approach [15] and with a RS-based approach where the mean imputation is used for the missing features before the RS ensemble classification. They show that Learn++.MF outperforms the compared methods across several datasets.

In this paper we compare several state-of-the-art approaches: EM, Learn++.MF, BPCA, and a variety of different machine learning imputation methods (each discussed below), and we propose a new method based on multiple imputation. The algorithms are tested on several different datasets. Furthermore, different state-of-the-art classifiers are compared with each coupled with an imputation method. We show that EM can also be coupled with random subspace, outperforming Learn++.MF. The simple fusion by sum rule between EM and our proposed approach obtains a very good performance even when 30% of features is missing.

Moreover, we have studied different combinations between standard imputation methods and our multiple imputation based on clustering showing that coupling clustering and a standard imputation method permits to improve the performance.

2. Compared systems

In this section we briefly report the state-of-the-art approaches that are used in our comparison experiments: mean imputation, artificial neural networks, InPaint, BPCA, kNN, EM, dissimilarity, and LearnMF.

Mean imputation (Mean) [16], which can be considered the simplest approach, uses the mean value of each non-missing variable to fill in missing values for all observations.

Artificial neural networks (NN) [28], is a machine learning procedure that creates a predictive model to estimate values that will substitute for the missing items. NN approaches model the missing data estimation using information available in the dataset. A NN estimates missing values by training a artificial neural network to learn the incomplete features (outputs), using the remaining complete features as inputs. In our experiments we have used the feed-forward backpropagation network as implemented in Mathworks MATLAB neural network toolbox, we have tested different parameters (learning epochs $\in \{100, 200, 400, 500\}$ and number of hidden nodes $\in \{3, 5, 9, 12\}$) and for each dataset we have reported the best result.

InPaint [18], replaces missing data by extra/interpolating the non-missing elements using an iterative process that converges toward the solution.[1]

BPCA [19], a Bayesian principal component analysis, is a probabilistic model and latent variables approach within the framework of Bayes inference. The missing value estimation method based on BPCA consists of three elementary processes: (1) principal component regression, (2) Bayesian estimation, and (3) a repetitive algorithm similar to expectation–maximization.[2]

kNN [20], is a method, where, given an incomplete pattern $\mathbf{x}$, K closest cases that are not missing values (i.e., features with missing values in x) in the features are imputed such that they minimize some distance measure. Once the K nearest neighbors have been found, a replacement value for the missing attribute value must be estimated. One obvious refinement is to weight the contribution of each neighbor according to its distance to $\mathbf{x}$.[3] We have tested different values for K ($K \in \{3, 5, 7, 9\}$ and selected the best value ($K = 3$)).

[1] Details and MATLAB code: http://www.mathworks.com/matlabcentral/fileexchange/27994-inpaint-over-missing-data-in-n-d-arrays (accessed: 08.10.2011).

[2] Details and MATLAB code: http://hawaii.sys.i.kyoto-u.ac.jp/~oba/tools/BPCAFill.html (accessed: 08.10.2011).

[3] Details and MATLAB code: http://www.mathworks.it/help/toolbox/bioinfo/ref/knnimpute.html (accessed: 08.10.2011).