ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Handling missing values in kernel methods with application to microbiology data

Vladimer Kobayashi[1] and Tomàs Aluja[2] and Lluís A. Belanche[3]

1- Laboratoire Hubert Curien - UMR CNRS 5516
Bâtiment F 18 Rue du Professeur Benoît Lauras
42000 Saint-Etienne FRANCE

2- Computer Science School - Dept. of Statistics & Operations Research
Technical University of Catalonia
Jordi Girona, 1-3 08034, Barcelona, SPAIN

3- Computer Science School - Dept. of Software
Technical University of Catalonia
Jordi Girona, 1-3 08034, Barcelona, SPAIN

Abstract. We discuss several approaches that make possible for kernel methods to deal with missing values. The first two are *extended* kernels able to handle missing values without data preprocessing methods. Another two methods are derived from a sophisticated *multiple imputation* technique involving logistic regression as local model learner. The performance of these approaches is compared using a binary data set that arises typically in microbiology (the microbial source tracking problem). Our results show that the kernel extensions demonstrate competitive performance in comparison with multiple imputation in terms of predictive accuracy. However, these results are achieved with a simpler and deterministic methodology and entail a much lower computational effort.

1 Introduction

Modern modelling problems are difficult for a number of reasons, including the challenge of dealing with a significant amount of missing information. Kernel methods have won great popularity as a reliable machine learning tool; in particular, Support Vector Machines (SVMs) are kernel-based methods that are used for tasks such as classification and regression, among others [1]. The kernel function is a very flexible container under which to express knowledge about the problem as well as to capture the meaningful relations in input space.

Some classical modelling methods –like Naïve Bayes and CART decision trees– are able to deal with missing values directly. However, the process of optimizing an SVM assumes that the training data set is complete. When present, missing values almost always represent a serious problem because they force to preprocess the dataset and a good deal of effort is normally put in this part of the modelling. In order to process such datasets with kernel methods, an imputation procedure is then deeemed a necessary but demanding step.

The aim of this paper is to examine and compare a number of approaches to handle missing values in kernel methods. Specifically, we present two methods that *extend* a kernel function in the presence of missing values and hence

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

handle missing values directly. We also investigate two different uses of the well-stablished multiple imputation method. These four approaches are used to analyze a fecal source pollution dataset presenting several challenges: it is a multi-class, small sample size problem plagued by missing values. All four have slightly better predictive accuracies than the best model suggested so far.

2 Preliminaries

Missing information is difficult to handle, specially when the lost parts are of significant size. Three possible ways to deal with missing data are: i) discard all observations (or variables) with missing values, ii) impute the values, and iii) extend the learner to work with incomplete observations. Deleting instances and/or variables containing missing values results in loss of relevant data and is also frustrating because of the effort in collecting the sacrificed information. Imputation methods entail inferring values for the missing entries [2, 3]. A growing number of studies recommend the use of multiple imputation –e.g. [4]. Compared to classical imputation, which imputes a single value, multiple imputation produces several values to fill the missing entries. These methods are independent of the learning algorithm and hence their impact on the learning process is uncertain. For SVMs, recent work tackles the problem by defining a modified risk that incorporates the uncertainty due to the missingness into a convex optimization task [5].

2.1 First kernel extension

The first kernel extension is obtained by wrapping a known kernel around a probability distribution [6]:

Theorem 1 *Let $\mathcal{X}$ denote a missing value. Let k be a kernel in X and P a probability mass function in X. Then the function $k_{\mathcal{X}}(x, y)$ given by*

$$k_{\mathcal{X}}(x,y) = \begin{cases} k(x,y), & \text{if } x, y \neq \mathcal{X} \ ; \\ \sum\limits_{y' \in X} P(y')k(x,y'), & \text{if } x \neq \mathcal{X} \text{ and } y = \mathcal{X}; \\ \sum\limits_{x' \in X} P(x')k(x',y), & \text{if } x = \mathcal{X} \text{ and } y \neq \mathcal{X}; \\ \sum\limits_{x' \in X} P(x') \sum\limits_{y' \in X} P(y')k(x',y'), & \text{if } x = y = \mathcal{X} \end{cases}$$

is a kernel in $X \cup \{\mathcal{X}\}$.

For binary variables $x, y \in \{0, 1\}$, define the kernel:

$$k(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \tag{1}$$

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Notice that this kernel is univariate. For the multivariate case $\mathbf{x}, \mathbf{y} \in X = \{0,1\}^d$, define the **first kernel extension** (**1KE**) as:

$$\mathcal{K}_1(\mathbf{x}, \mathbf{y}) = \frac{1}{d} \sum_{i=1}^{d} k_{\mathcal{X}}(x_i, y_i) \tag{2}$$

2.2 Second kernel extension

The **second kernel extension** (**2KE**) deals with the multivariate case directly but is limited by the number of variables. The idea is to consider all possible *completions* of an observation with missing values. An example will prove helpful: suppose we have an incomplete observation given by $(0, 1, \mathcal{X}, 0)$; then the possible completions for this observation are $\{(0,1,0,0), (0,1,1,0)\}$.

Theorem 2 *Let $\mathcal{X}$ denote a missing value. Let k be a kernel in $X = \{0,1\}^d$. Let $c(\mathbf{x})$ be the set of completions of $\mathbf{x}$. Given two vectors $\mathbf{x}, \mathbf{y} \in X$, the function*

$$\mathcal{K}_2(\mathbf{x}, \mathbf{y}) = \frac{1}{|c(\mathbf{x})||c(\mathbf{y})|} \sum_{\mathbf{x}' \in c(\mathbf{x})} \sum_{\mathbf{y}' \in c(\mathbf{y})} k(\mathbf{x}', \mathbf{y}') \tag{3}$$

is a kernel in $X \cup \{\mathcal{X}\}$.

Proof. The set of kernels is a convex cone; therefore it is closed under linear combinations with positive coefficients.

2.3 Multiple imputation methods

These methods involve the estimation of what the missing values could have been and then use the completed datasets for modelling. Two main methods for multivariate data have been proposed: *joint modeling* (**JM**) and *fully conditional specification* (**FCS**). **JM** assumes a (multivariate) distribution for the missing data, and draws imputed values from the conditional distributions by MCMC techniques. **JM** techniques are available if the distribution is assumed to be multivariate normal, log-linear or a general location model. The success of **JM** depends on the impact of these assumptions. On the other hand, **FCS** does not make distributional assumptions in advance, since it specifies a multivariate imputation model on a variable-by-variable basis by a set of conditional densities, one for each incomplete variable [7]. **FCS** will start with an initial imputation and then draw imputations by iterating over the conditional densities [8].

Let us denote our observation as $X = (X_1, \ldots, X_d)$, possibly with missing values. The observed and missing parts of X are denoted by X^{obs} and X^{mis}, respectively. Let $X_{-j} = (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_d)$ denote the collection of the $d-1$ variables in X except X_j.

The hypothetically complete observation X is assumed to be drawn from a d-variate distribution $P(X|\boldsymbol{\theta})$. We assume that the multivariate distribution of

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

X is completely specified by $\boldsymbol{\theta}$, a vector of unknown parameters. The parameters $\theta_1, \ldots, \theta_d$ are specific to the respective conditional densities and are not necessarily the product of a factorization of the *true* joint distribution $P(X|\boldsymbol{\theta})$.

The *chained equations* method obtains the posterior distribution for $\boldsymbol{\theta}$ by sampling iteratively from conditional distributions of the form $P(X_j|X_{-j}, \theta_j), j = 1, \ldots, d$. Starting from a simple draw from the observed marginal distributions, the t-th iteration of the process is a *Gibbs sampler* to successively draw [8]:

$$\begin{aligned}
\theta_1^{*(t)} &\sim P(\theta_1 \,|\, X_1^{\text{obs}}, X_2^{(t-1)}, \ldots, X_d^{(t-1)}) \\
X_1^{*(t)} &\sim P(X_1^{\text{mis}} \,|\, X_1^{\text{obs}}, X_2^{(t-1)}, \ldots, X_d^{(t-1)}, \theta_1^{*(t)}) \\
&\vdots \\
\theta_d^{*(t)} &\sim P(\theta_d \,|\, X_d^{\text{obs}}, X_1^{(t)}, \ldots, X_{d-1}^{(t)}) \\
X_d^{*(t)} &\sim P(X_d^{\text{mis}} \,|\, X_d^{\text{obs}}, X_1^{(t)}, \ldots, X_d^{(t)}, \theta_d^{*(t)})
\end{aligned}$$

where $X_j^{(t)} = (X_j^{\text{obs}}, X_j^{*(t)})$ is the jth imputed variable at iteration t. Observe that previous imputations $X_j^{*(t-1)}$ only enter $X_j^{*(t)}$ through its relation with other variables, and not directly. Moreover, and unlike other MCMC methods, no information about X_j^{mis} is used to draw $\theta_j^{*(t)}$, so convergence can be quite fast. The procedure is iterated a number of m times to generate m different multiple imputations. For the technique to be practical, a univariate imputation model is needed for each of the incomplete variables. The choice will be steered by the scale of the dependent variable (the variable that we need to impute), and preferably incorporates knowledge about the relation between the variables. Other considerations include the set of variables to include as predictors; the order in which variables should be imputed; the number of imputed data sets; whether we should impute variables that are functions of other (incomplete) variables; the form of the starting imputations and the number of iterations.

3 Experimental evaluation

3.1 Problem description

The study of fecal source pollution in waterbodies is a major problem in ensuring the welfare of human populations, given its incidence in a variety of diseases, specially in under-developed countries. Microbial source tracking methods attempt to identify the source of contamination, allowing for improved risk analysis and better water management [9]. The available data set includes a number of chemical, microbial, and eukaryotic markers of fecal pollution in water. All variables (except the class variable) are binary, i.e., they signal the presence or absence of a particular marker. The data set includes 9 binary variables, 138 observations and four classes, with 212 missing entries out of $1,242$ (approximately 17%). All variables have percentages of missing entries greater than 15%. A recent study

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

investigating this data set reported a Naïve Bayes classifier as the best model, yielding an accuracy of 77.9% [10].

3.2 Performing multiple imputation

To our knowledge there has been little work in using multiple imputation for missing data treatment prior to the application of a SVM as the learning algorithm. The crucial element is how to pool the results coming from several SVMs that are trained for each imputed data set. In this paper we propose two methods to do this pooling. The first method is to concatenate the multiply imputed data sets and optimize an SVM classifier in the resulting set; this not only accounts for the variability of the parameter estimates but also for the variability of the training observations in relation to the imputed values. The second, more standard procedure, involves fitting separate SVMs to each imputed data set and get the average performance of the different SVMs. Since the missing values in our problem are found in variables which are binary, logistic regression is a good choice for the imputation models. One is also required to identify which of the remaining variables will be used as predictors. To this end we compute the *Kendall* rank correlation coefficient for each pair of variables and set a threshold that will serve as an indicator for a variable to be included as a predictor. In addition, we also determine the *proportion of usable cases* (PUC); this will tell us whether a predictor contains only fractional information to impute the target variable, and thus could be dropped from the model. To improve the imputation model we decided to use the class variable as a predictor whenever it is appropriate (as indicated by the *Kendall* coefficient and the PUC). Finally, the number of imputed data sets is set to 10, to keep computations manageable.

3.3 Results and discussion

Four separate predictive models were built for each of the approaches: **1KE**, **2KE**, the first version of multiple imputation (**1MI**) and the second version of multiple imputation (**2MI**)[1]. To obtain a reliable estimate of predictive accuracy in this small data set, a stratified 10 times 10-fold cross-validation (10x10cv) is performed. Table 1 summarizes the results.

			10x10cv for each class			
Approach	C	10x10cv	Human	Cow	Poultry	Swine
1KE	2.0	79.3	95.4	64.5	75.2	69.4
2KE	1.6	78.2	92.6	62.8	71.8	74.2
1MI	1.0	79.9	92.7	66.4	69.4	80.2
2MI	1.0	79.0	94.5	57.5	70.8	78.8

Table 1: Mean 10x10cv accuracies for the four approaches to handle missing values. Also shown are best cost parameter C and detailed class performance.

[1]We used the `R` software [11], extended with the `kernlab` package (for the SVMs) and the `mice` package, which implements the **FCS** method for multivariate multiple imputation.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

The table shows that all four approaches have comparable performance, with **2KE** seeming inferior; **1KE** performs best in identifying *human* contamination, the most important single decision; the two multiple imputations are good for *swine* origin. The four approaches have difficulties in classifying *cows*, probably because this class is the minority class, representing only 18% of the observations.

4 Conclusions

In real problems, accuracy may not tell the whole picture about a model. Other performance criteria include development cost, interpretability, and utility. The *cost* here refers to how much pre-processing effort and computing time we need in order to build the model. Undoubtedly, the two imputation methods require more time and resources compared to the kernel extensions. Prior to imputation a good univariate imputation model must be identified for each variable containing missing values. These methods depend also on several non-trivial algorithmic options and have an added computational cost for training separate SVMs for each of the imputed data sets. *Interpretability* refers to the complexity of the obtained model. It is unclear if a model that had values imputed (several times) is more interpretable than one that had not. Finally, the model must be *useful* in practice: in a real deployment of the model, new and unseen observations emerge which we need to classify, which may contain missing values. The two kernel extensions are the only methods able to face this situation.

References

[1] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge Univ. Press, 2004.

[2] R. Little and D. Rubin. *Statistical Analysis with Missing Data*. Wiley-Interscience, 2009.

[3] Q. Song and M. Shepperd. Missing data imputation techniques. *Int. J. Business Intelligence and Data Mining*, 2(3):261-291, 2007.

[4] Y. He. Missing data analysis using multiple imputation. *Circulation: Cardiovascular Quality and Outcomes*, 3(1):98–105, 2010.

[5] K. Pelckmans, J. De Brabanter, J. Suykens and B. De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18:684-692, 2005.

[6] G. Nebot and Ll. Belanche. A kernel extension to handle missing data. In Bramer, Ellis and Petridis, editors, *Research and Development in Intelligent Systems XXVI*, Springer, 2010.

[7] K. Lee and J. Carlin. Multiple imputation for missing data: Fully conditional specification vs multivariate normal imputation. *Amer. Journal of Epidemiology*, 171(5):624-632, 2010.

[8] S. van Buuren and K. Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 2011.

[9] T.M. Scott, J.B. Rose, T.M. Jenkins, S.R. Farrah and J. Lukasik. Microbial source tracking: Current methodology and future directions. *Applied and Environmental Microbiology*, 68(12):5796–5803, 2002.

[10] E. Ballesté, X. Bonjoch, Ll. Belanche and A. R. Blanch. Molecular indicators used in the development of predictive models for microbial source tracking. *Applied and Environmental Microbiology*, 76(6):1789–1795, 2010.

[11] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008.