Naïve Bayes vs. Support Vector Machine: Resilience to Missing Data

Hongbo Shi and Yaqin Liu

School of Information Management, Shanxi University of Finance and Economics
030031 Taiyuan, China
shb710@163.com, liuyaqin2003@126.com

Abstract. The naïve Bayes and support vector machine are the typical generative and discriminative classification models respectively, which are two popular classification approaches. Few studies have been done comparing their resilience to missing data. This paper provides an experimental comparison of the naïve Bayes and support vector machine regarding the resilience to missing data on 24 UCI data sets. The experimental results show that when the missing rate is very small (e.g. 1%), the resilience of the naïve Bayes classifiers to missing data are approximately similar to that of support vector machine classifiers. With the increase of the missing rate, however, the resilience of the naïve Bayes classifiers to missing data are slowly decreased and that of support vector machine classifiers to missing data are rapidly decreased. This demonstrates that the naïve Bayes classifiers have better resilience to missing data than support vector machine classifiers.

Keywords: missing data, the naïve Bayes, SVM, resilience.

1 Introduction

Missing data is a common problem that appears in many real world situations. For example, sensor failures in industrial control processes, omitted entries in databases and non-response in questionnaires [1]. Many scientific, industrial, business and economic decisions are related to the information available at the time of making decisions. In these applications, if we merely ignore the incomplete instance or handle inappropriately missing values, it may lead to biased results in statistical modeling. Therefore, it is essential to research on the problem of missing data.

Many researchers engaged in a serious study of missing data. In order to identify the reason why data are missing, Little and Rubin define three different types of missing data mechanisms [2]: missing completely at random, missing at random and not missing at random. To take advantage of missing data, some common methods handling missing data, which used before learning algorithms, are proposed, for example, case deletion, attribute deletion, mean imputation, multiple imputation and so on. The most representative classification algorithms which are able to deal with missing values were investigated, such as decision trees[3], fuzzy approaches[4], Bayes approaches[5] and support vector machines[6]. In addition, [7] examined the

H. Deng et al. (Eds.): AICI 2011, Part II, LNAI 7003, pp. 680–687, 2011.

effect of missing data to different classification algorithms, including two rule inducers, a nearest neighbor method, two decision tree inducers, a naïve Bayes inducer, and linear discriminant analysis. They found that the naïve Bayes method was by far most resilient to missing data.

Generative and discriminative approaches are two different paradigms for solving classification problems, which have different thoughts and frameworks. The discriminative approaches look for an optimal decision function $f(\boldsymbol{x})$ or the probability $p(y|\boldsymbol{x})$ of $\boldsymbol{x}$ being the class y to separate the data from data with the other class label, whereas a generative model often captures the generation process of $\boldsymbol{x}$ by modeling $p(\boldsymbol{x}|y)$ and tries to represent the true density of the data. The naïve Bayes classifier and support vector machine (SVM) are the typical generative and discriminative models, respectively. In this paper, we compare the naïve Bayes with support vector machine for examining their reliance to missing data. We select these two particular algorithms for several reasons. First, they are popular with data analysts, machine learning researchers, and statisticians. Second, the naïve Bayes and support vector machine are the generative and discriminative approach, respectively. Third, they often are applied to handle higher dimension data, for instance, text data.

2 Naïve Bayes Classifier vs. Support Vector Machine Classifier

2.1 Naive Bayes Classifier

The naïve Bayes classifier is a typical generative classifier, which can be regarded as a special case of Bayesian network classifiers [8]. In general, Bayesian network classifier models first the joint distribution $p(\boldsymbol{x},y)$ of the measured attributes $\boldsymbol{x}$ and the class labels y factorized in the form $p(\boldsymbol{x}|y)p(y)$, and then learns the parameters of the model through maximization of the likelihood given by $p(\boldsymbol{x}|y)p(y)$. Due to there is a fundamental assumption that the attributes are conditionally independent given a target class, the naïve Bayes classifier in fact learns the parameters of the model through maximization of the likelihood given by $p(y)\prod_j p(x_j|y)$.

Since the naïve Bayes classifiers optimize the model over the whole dimensionality, and are capable of learning even in the presence of some missing values. Furthermore, the naïve Bayes classifier is a stable, and its classification result is not significant changed due to noises or corrupted data.

2.2 SVM Classifier

The SVM [9] classifier is a typical discriminative classifier. Different from generative classifier, it mainly focuses on how well they can separate the positives from the negatives, and does not try to understand the basic information of the individual classes. The SVM classifier maps first the instance $\boldsymbol{x}$ in a training set into a high dimensional space via a function Φ, then computes a decision function of the form $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b$ by maximizing the distance between the set of points $\Phi(\boldsymbol{x})$ to the hyperplane or set of hyperplanes parameterized by $(\boldsymbol{w}, b)$ while being consistent on the training set.

The SVM classifier builds a single model for all classes and hence it requires simultaneous consideration of all other classes. Moreover, the SVM classifier

performs well on data sets that have many attributes, even if there are very few cases on which to train the model.

3 Missing Data Mechanisms

For data sets with missing values $\boldsymbol{X}=\{\boldsymbol{x}^{(i)}, y^{(i)} \mid i=1,\ldots,n\}$, all attribute values can be partitioned into two categories: observed values and missing values, and hence each instance $\boldsymbol{x}^{(i)}$ is composed of two parts: the observed attribute values $\boldsymbol{x}_{\boldsymbol{o}_i}^{(i)}=\{x_j^{(i)} \mid j \in \boldsymbol{o}_i\}$ and the missing attribute values $\boldsymbol{x}_{\boldsymbol{m}_i}^{(i)}=\{x_j^{(i)} \mid j \in \boldsymbol{m}_i\}$, where $\boldsymbol{o}_i$ and $\boldsymbol{m}_i$ are the set of indices for observed and missing attributes, respectively. Each instance $\boldsymbol{x}^{(i)}$ has its own observed set $\boldsymbol{o}_i$ and missing set $\boldsymbol{m}_i$.

Let $\boldsymbol{M}=\{M_j^{(i)} \mid i=1,\ldots,n,\ j \in \boldsymbol{o}_i \cup \boldsymbol{m}_i\}$ be a missing data indicator matrix. If $x_j^{(i)}$ is observed, then $M_j^{(i)}=1$, otherwise, $M_j^{(i)}=0$. We refer to $\boldsymbol{M}$ as the *missingness*. The parameters of characterizing the distribution of *missingness* is usually called the missing data mechanism ξ, therefore the missing data mechanism is characterized by the conditional distribution of $\boldsymbol{M}$ given the input data set $\boldsymbol{X}=(\boldsymbol{X}_m, \boldsymbol{X}_o)$,

$$p(\boldsymbol{M} \mid \boldsymbol{X}, \xi) = p(\boldsymbol{M} \mid \boldsymbol{X}_m, \boldsymbol{X}_o, \xi) \tag{1}$$

where $\boldsymbol{X}_o$ and $\boldsymbol{X}_m$ are the observed input set and the unknown input set, respectively.

According to whether there is the dependence relationship between the *missingness* and data, [2] defines three types of the missing data mechanisms: missing completely at random (MCAR), missing at random (MAR) and not missing at random (NMAR).

- MCAR

In MCAR situation, there are no constraints to the relationship between the *missingness* and data, and the missing values are randomly distributed across all observations, and the probability of *missingness* does not depend on the values of other covariates. The MCAR condition can be expressed by the relation

$$p(\boldsymbol{M} \mid \boldsymbol{X}_m, \boldsymbol{X}_o, \xi) = p(\boldsymbol{M} \mid \xi) \tag{2}$$

Since there are not dependence between missing values and other data values in MCAR situation, the basic way of handing missing data, e.g. mean imputation or special values imputation, may be valid.

- MAR

For MAR, the missing values are not randomly distributed across all observations, the probability of the *missingness* is conditional on the values of other covariates, and the missing values are therefore randomly distributed within subsets of observations.

The *missingness* is independent of the missing variables but the pattern of data *missingness* is traceable or predictable from other variables in the database. The MAR condition can be expressed by the relation

$$p(\boldsymbol{M} \mid \boldsymbol{X}_m, \boldsymbol{X}_o, \xi) = p(\boldsymbol{M} \mid \boldsymbol{X}_o, \xi) \tag{3}$$

- NMAR

In NMAR situation, the missing values are not distributed randomly. In contrast to the MAR and MCAR situation, the probability of the *missingness* cannot be predicted

from the values of other covariates and may depend on the values of the missing data. If the probability of the *missingness* can not be modeled, it is likely to lead to biased model estimates. There is a no general method of handling missing data properly.

When data are MCAR or MAR, the missing data mechanism is known as ignorable. Ignorable mechanisms are important, because when they occur, a researcher can ignore the reasons for missing data in the analysis of the data, and thus simplify the methods used for missing data analysis [10]. Therefore, the majority of research works focuses on the MAR or the MCAR situation.

4 Experimental Setup and Results

4.1 Data Sets

In order to investigate the properties of the Naïve Bayes and support vector machine classifier with respect to missing values, we chose 24 data sets from the UCI machine learning repository [11]. Since the main goal of the experiments is to explore the impact of missing rate on different classification algorithms, all of data sets have not missing values.

The MCAR was chose as the missing data mechanism of the experiments. To simulate the missing completely at random setting, we randomly replaced a fraction of attribute values with '?' ('?' represents missing values) according to a uniform distribution, and assumed the rest are observed. Following the method of [7], the proportion of missing data ranges from 1% to 40%, i.e. 1%, 5%, 10%, 15%, 20%, 25%, 30%, 35% and 40%. On each original data set, nine data sets with different missing rate were generated by our data generating algorithm.

4.2 Evaluation Metrics

Before starting all of the experiments, we need to define a performance measure that is appropriate for measuring the impact of missing data. The classification accuracy has been the standard comparison metric used in studies of classifier induction in the machine learning literature, which is the number of correct predictions on the test data divided by the number of test data instances. Since accuracy can be affected by the data structure and by the missing data, and our focus is to compare the resilience of various classification algorithms to missing data, we need to define a measure called resilience of missing data (*ResMiss*), calculated as

$$ResMiss = \frac{Accuracy_{missing} - Accuracy_{original}}{Accuracy_{original}} \times 100\% \tag{4}$$

where $Accuracy_{missing}$ and $Accuracy_{original}$ represent accuracy with missing data and original full data, respectively. The *ResMiss* measures the difference between the accuracy achievable with missing values and achievable with the original full data relative to that achievable with the original full data. The smaller the *ResMiss* is, the better the resilience of an algorithm to missing data is. If *ResMiss* of an algorithm is equal to 0, the algorithm will not affected by the existing missing values in the data set.

4.3 Workbench and Methodology

All the experiments were performed in the Weka system [12], which provides a workbench that includes full and working implementations of many popular learning schemes. To compare the resilience of the naïve Bayes to missing data with that of support vector machine, we need first to obtain their classification accuracy on different data sets. We use "weka.classifiers.bayes.NaiveBayes" as the naïve Bayes classifier (simply called NB) and "weka. classifiers.functions.SMO" as the support vector machine classifier (simple called SVM) implemented in Weka system.

The classification accuracy is observed via 10 runs of 10-folds stratified cross validation on data sets. $Accuracy_{missing}$ and $Accuracy_{original}$ are obtained on missing data sets and the original data sets, respectively. Then, we compute *ResMiss* of the naïve Bayes classifier and SVM classifier by using the equation (4).

4.4 Result and Analysis

Fig. 1 shows that the comparison of the average *ResMiss* of the naïve Bayes and support vector machine to missing data on 24 data sets. The X axis is the percentage of missing values with respect to the original data set, and the Y axis is the measure *ResMiss* of the resilience of missing data.

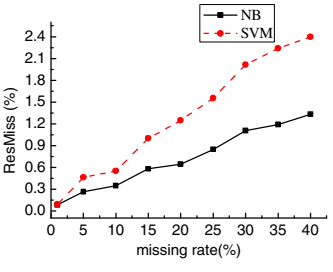

Fig. 1. Comparison of the average ResMiss of the naïve Bayes classifier with that of the SVM classifier with repect to missing data on 24 data sets

From Fig. 1, we can obviously observe that when the missing rate is about 1%, the average *ResMiss* of the naïve Bayes and the SVM are approximately the same. With the increase of the missing rate, the average *ResMiss* of the naïve Bayes classifier goes up in relatively slow speed and the average *ResMiss* of the SVM goes rapidly up. When the missing rate is up to 40%, the average *ResMiss* of the naïve Bayes is 1.2% lower than that of SVM. That is to say, only when the missing rate is very small, the resilience of the naïve Bayes and the SVM are approximately the same. In general, the resilience of the naïve Bayes classifier to missing data is superior to that of the SVM classifier to missing data, especially in the case that the missing rate is higher.

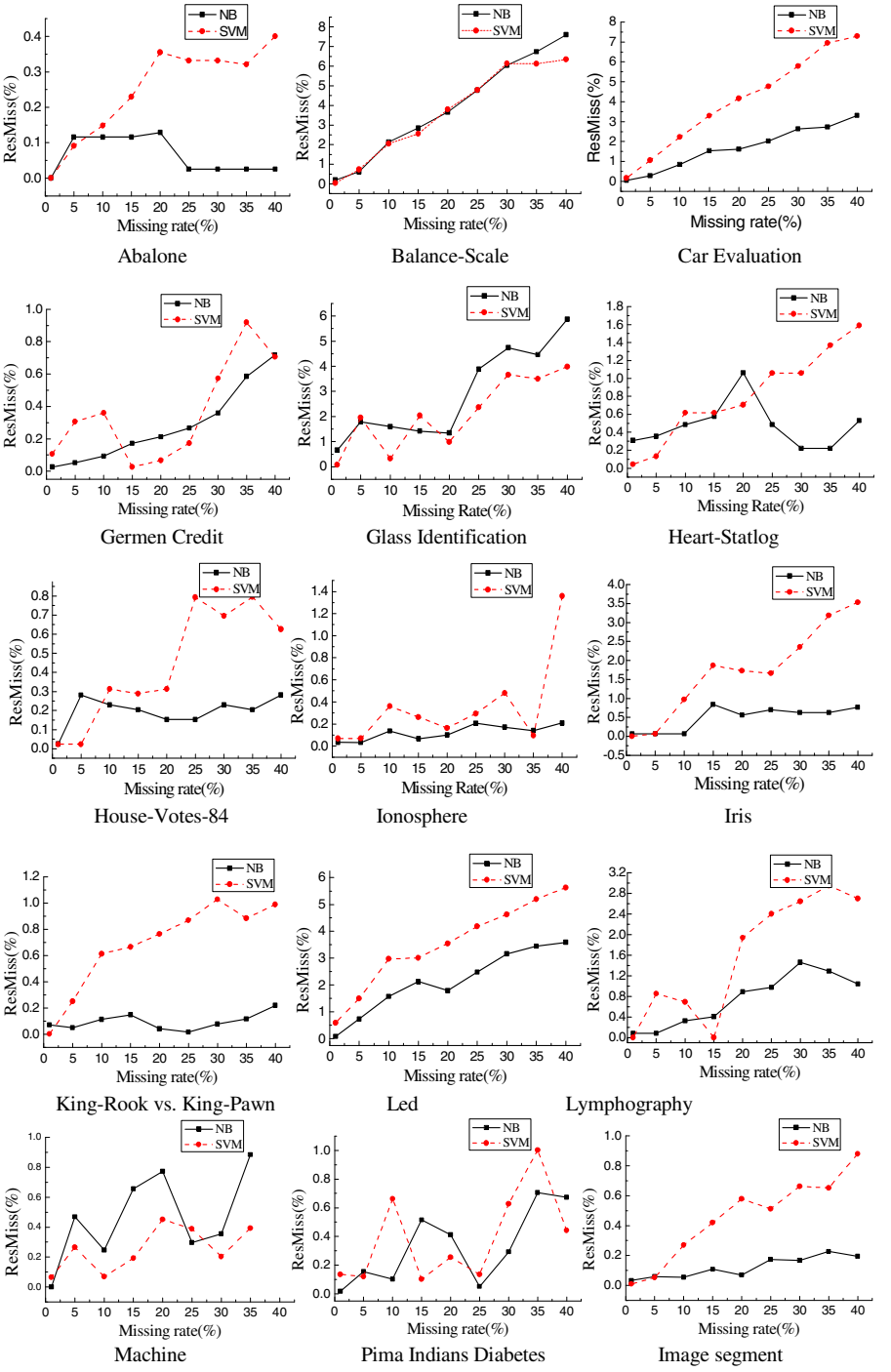

Fig. 2. Comparison of the resilience of the naïve Bayes classifier and the SVM classifier to missing data on 24 data sets

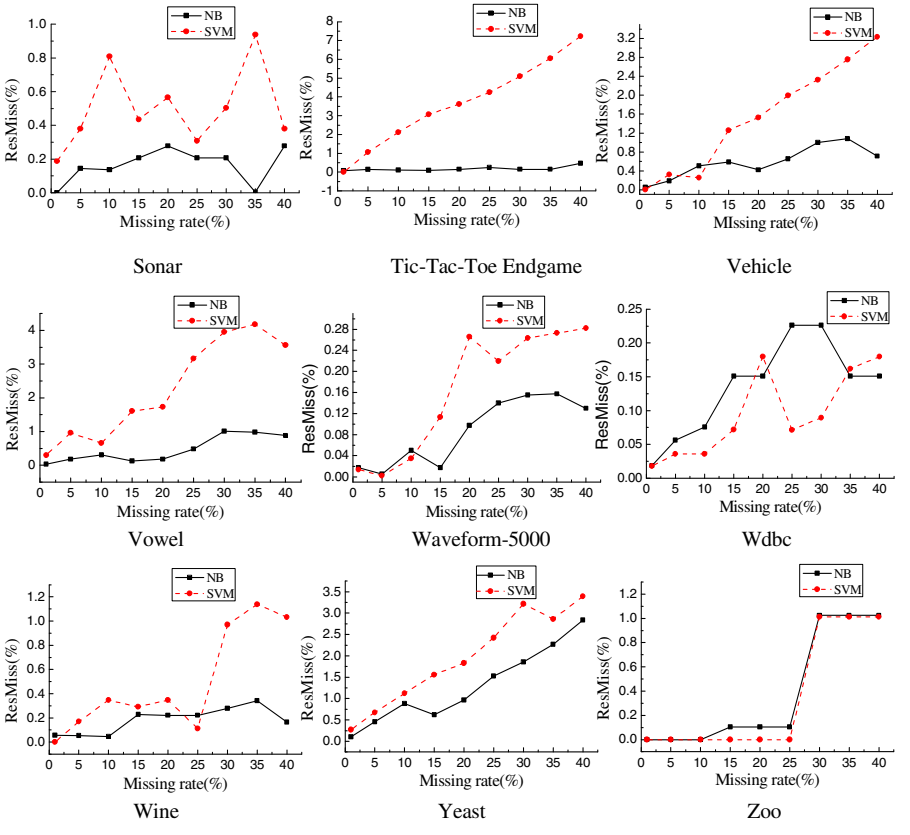

Fig. 2. (*continued*)

To compare clearly the resilience of the naïve Bayes with that of the SVM on each data set, Fig. 2 shows *ResMiss* comparison of the naïve Bayes and the SVM on 24 data sets. From Fig. 2, some results can be observed:

- For most data sets, the resilience of the Naïve Bayes classifier to missing data is superior to that of the SVM.
- Merely for individual data sets, the above conclusion is not true. For wdbc, machine and glass, the resilience of the Naïve Bayes classifier to missing data is inferior to that of the SVM; for blance-scale and zoo, the resilience of the Naïve Bayes classifier to missing data is very similar to that of the SVM.
- For all of data sets, when the missing rate is very small, the naïve Bayes and the SVM are almost same resilient to missing data. With the increase of the the missing rate, however, the gap of the resilience to missing data between the naïve Bayes and the SVM is significantly increased.

5 Conclusions

The naïve Bayes and support vector machine are the typical generative and discriminative classification model, respectively. To understand the effect of missing data to two classification approaches, this paper conducted an experimental comparison of the naïve Bayes classifiers and the support vector machine classifiers regarding their resilience to missing data. The experiments were performed on 24 UCI data sets. The experimental results show that the naïve Bayes classifiers have better resilience to missing data than the support vector machine classifiers.

Our experimental results are obtained in MCAR situation, and whether these conclusions are true in MAR and NMAR situation need further to investigate.

Acknowledgments. This paper is funded by the National Natural Science Foundation of China under Grant No. 60873100 and the Natural Science Foundation of Shanxi Province of China under Grant No. 2009011017-4 and No. 2010011022-1.

References

1. García-Laencina, P.J., Sancho-Gómez, J.L., Figueiras-Vidal, A.R.: Pattern classification with missing data: a review. Neural Computation & Applications 9, 1–12 (2010)
2. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data, 2nd edn. John Wiley & Sons, New York (2002)
3. Webb, G.I.: The problem of missing values in decision tree grafting. In: 10th Australian Joint Conference on Artificial Intelligence, pp. 273–283. Springer, London (1998)
4. Ichihashi, H., Honda, K.: Fuzzy c-means classifier for incomplete data sets with outliers and missing values. In: International Conference on Computational Intelligence for Modeling, Control and Automation, pp. 457–464. IEEE Computer Society, Washington, DC (2005)
5. Ramoni, M., Sebastiani, P.: Robust Bayes classifier. Artificial Intelligence 125, 209–226 (2001)
6. Pelckmans, K., Brabanter, J.D., Suykens, J.A.K., Moor, B.D.: Handling missing values in support vector machine classifiers. Neural Network 18, 684–692 (2005)
7. Kalousis, A., Hilario, M.: Supervised knowledge discovery from incomplete data. In: 2nd International Conference on Data Mining. WIT Press, Cambridge (2000)
8. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29, 131–163 (1997)
9. Vapnik, V.: Statistical learning theory. John Wiley & Sons, New York (1998)
10. Schafer, J.L.: Analysis of incomplete multivariate data. Chapman & Hall, Florida (1997)
11. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2010), `http://archive.ics.uci.edu/ml`
12. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, Seattle (2000)