International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume3, Issue 6 (June '2017)

MODIFING INCONSISTENT, NOISY DATA USING NB-EM ALGORITHM

M.Devaraj,
M.Phil- Research Scholar,
Department of Computer Science
AJK College Of Arts and Science
Coimbatore,Tamilnadu,India.

N. Santhana Krishna,
Head,
Department of Information Technology,
AJK College Of Arts and Science
Coimbatore,Tamilnadu,India.

Abstract: In data mining applications, there are various kinds of missing values in experimental datasets. Non-substitution or inappropriate treatment of missing values has a high probability to cause a lot of warnings or errors. Besides, many classification algorithms are very sensitive to the missing values. Because of these, handling the missing values is an important phase in many classification or data mining task. This paper introduces traditional EM algorithm and disadvantage of the EM algorithm. This paper proposes a new method to implement the missing values based on EM algorithm, which uses Naive Bayesian to improve the accuracy. We conclude by classifying seeds dataset and vertebral columns dataset and comparing the results to those obtained by applying two other missing value handling methods: the traditional EM algorithm and the non-substitution method. The experimental results prove a stable algorithm for improving the data classification accuracy on large datasets, which contain a lot of missing values.

Keywords: Data mining, Data cleaning, Expectation Maximization Algorithm

I.INTRODUCTION

Data cleaning is part of data preprocessing before data mining, prior to process of mining information in a data warehouse, data cleaning is crucial because of garbage in and garbage out principle. Data cleaning is also called data cleansing or scrubbing deals with detecting and removing errors and inconsistencies from data in order to improve quality of data. Missing values exist in many situations wherein no values are reserved for some variable in an experiment or observation. In real-life data, some stored values are frequently missing as a result of unexpected mistakes, most often because either they are lost or they are independent of conditions. Although missing values are a common occurrence, they can nonetheless have a significant effect on the processing of and results derived from data. The missing values frequently infect the operating performance and produce mistakes in the mining model some classification algorithms, such as back propagation neural network, K-Nearest neighbor algorithm, C4.5 decision trees and so on, are very sensitive to the missing values. If there are a lot of missing values in datasets, then use one of among algorithms to classify, user will have a high probability to obtain the low classification accuracy. Accordingly, handling missing values is an important step in preprocessing phases for most data classification or data mining tasks. Inappropriate implementation of missing values can produce serious errors or false results.

Generally, methods for dealing with missing values can be divided into three classes: i) Delete the missing values, ii) Implement the missing values with the estimated values, and iii) ignore the missing values. Among these methods, deleting missing values is the easiest. However, when the rate of missing values in each attribute is high, this method has an unsatisfactory performance. Ignoring missing values also causes similar issues. Thus naturally prefer methods for implementing missing values. There are many methods for accomplishing this, such as the approximation, stochastic regression, and neural network methods. Among all the approaches, the EM (expectation-maximization) algorithm can reliably use the stable and the maximum step to find the optimal values for implementing the missing values. There are many methods for accomplishing this, such as the approximation, stochastic regression, and neural network methods. Among all the approaches, the EM (expectation-maximization) algorithm can reliably use the stable and the maximum step to find the optimal values for implementing the missing values. However, the EM algorithm's speed of convergence is quite slow and easily falls into local optimization. Let if fixed initial value given, it increases the speed of convergence and algorithm stability. The main objective of data cleaning is to reduce the time and complexity of mining process and increase the quality of data in data warehouse. The quality of data can be increased by using data cleaning techniques. To be processable and interpretable in a effective and efficient manner, data has to satisfy a set of quality criteria. Data satisfying those quality criteria is said to be high quality. At the same time, the user can overcome the deviation by way of marginal values. Together, these give the EM algorithm a better performance. This improved EM algorithm is based on Naive Bayesian, and named the NB-EM algorithm, which uses the result of classification to substitute otherwise-random initial values.

II.DATA CLEANING TECHNIQUES

Real world data tend to be in complete, noisy and inconsistent. Data cleaning routines attempts to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

a) Ways for handling missing values

- **Ignore the tuple:** This is usually done when class label is missing. This method is not very effective, unless tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attributes varies considerably.

- **Fill in the missing value manually:** This approach is time consuming and may not be feasible given a large data set with missing values.
- **Use a global constant to fill in the missing value:** Replace all missing attribute values by the same constant, such as label like "unknown". if missing value are replaced by unknown then the mining program may mistakenly think that they form an interesting concept ,since they all have a value common – that of "unknown".
- **Use the attribute mean to fill in the missing value:** For example, suppose that the average income of AllElectronics customers is Rs.28,000. Use this value to replace the missing value for income.
- **Use the attribute mean for all samples belonging to the same class as the given tuple:** For example, if classifying customers according to credit risk, replace the missing value with the average income value for customers in the same credit risk category as that of the given tuple.
- **Use the most probable value to fill in the missing value:** This may be determined with regression, inference-based using tools.

b) Noisy data

Noise is a random error or variance in a measured variable. Given a numeric attribute such as, price, etc,. The following methods describe to handle such kind of data.

- **Binning methods**: Binning methods smooth a sorted data value by consulting the "neighborhood", or values around it. The sorted values are distributed into a number of "buckets", or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. Figure illustrates some binning techniques. In this example, the data for price are first sorted and then partitioned into equal-frequency bins of size 3 (i.e., each bin contains 3 values). In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.
- **Clustering:** Outliers may be detected by clustering, where similar values are organized into groups or "clusters". Intuitively, values which fall outside of the set of clusters may be considered outliers.
- **Regression**: Data can be smoothed by fitting the data to a function, such as with regression. Linear regression involves finding the "best" line to fit two variables, so that one variable can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

c) Inconsistent data

There may be inconsistencies in the data recorded for some transactions. Some data inconsistencies may be corrected manually using external references. For example, errors made at data entry may be corrected by performing a paper trace. This may be coupled with routines designed to help correct the inconsistent use of codes. Knowledge engineering tools may also be used to detect the violation of known data constraints. For example, known functional dependencies between attributes used to find values contradicting the functional constraints.

III.EM Algorithm

The EM algorithm is a popular method of iterative refinement. In each iterative step, it has an Expectation Step and a Maximization Step, where the Expectation Step estimates the missing values and the Maximization Step updates the model parameters. The basis of the algorithm is to first estimate the missing value's initial values and obtain the values of the model parameters, and then to iteratively repeat the Expectation Step and Maximization Step, while updating the estimated values, until the function reaches convergence.

In more detail:

(1) Randomly choose K samples as the center of each class.

(2) Repeat Expectation Step and Maximization Step to improve the accuracy, until the function reaches convergence. a. Expectation Step: use the probability ( ) k P X ∈C i to classify each sample into some Class k.

Suppose let augment with the latent variable z that indicates which of the k Gaussians our observation y came from. The derivation of the E and M steps are the same as for the toy example, only with more algebra.

For the M-step,

Find $\theta=(w,\mu,\Sigma)\theta=(w,\mu,\Sigma)$ that maximizes Q. By taking derivatives with respect to $(w,\mu,\Sigma)(w,\mu,\Sigma)$ respectively and solving (remember to use Lagrange multipliers for the constraint that $\sum_{j=1}^{k} wj = 1$, we get

```
from scipy.stats import multivariate_normal as mvn
def em_gmm_orig(xs, pis, mus, sigmas, tol=0.01, max_iter=100):
    n, p = xs.shape
    k = len(pis)
    ll_old = 0
    for i in range(max_iter):
        exp_A = [ ]
        exp_B = [ ]
        ll_new = 0
        # E-step
        ws = np.zeros((k, n))
        for j in range(len(mus)):
            for i in range(n):
                ws[j, i] = pis[j] * mvn(mus[j], sigmas[j]).pdf(xs[i])
        ws /= ws.sum(0)
    # M-step
        pis = np.zeros(k)
        for j in range(len(mus)):
            for i in range(n):
                pis[j] += ws[j, i]
        pis /= n
        mus = np.zeros((k, p))
        for j in range(k):
            for i in range(n):
                mus[j] += ws[j, i] * xs[i]
            mus[j] /= ws[j, :].sum()
        sigmas = np.zeros((k, p, p))
        for j in range(k):
            for i in range(n):
                ys = np.reshape(xs[i]- mus[j], (2,1))
                sigmas[j] += ws[j, i] * np.dot(ys, ys.T)
            sigmas[j] /= ws[j,:].sum()
        # update complete log likelihoood
```

```
ll_new = 0.0
for i in range(n):
    s = 0
    for j in range(k):
        s += pis[j] * mvn(mus[j], sigmas[j]).pdf(xs[i])
    ll_new += np.log(s)
if np.abs(ll_new - ll_old) < tol:
    break
ll_old = ll_new
 return ll_new, pis, mus, sigmas
```

IV.REPLACE MISSING VALUES WITH EM ALGORITHM BASED ON GMM AND NAÏVE BAYESIAN

In data mining applications, there are various kinds of missing values in experimental datasets. Non-substitution or inappropriate treatment of missing values has a high probability to cause a lot of warnings or errors. Besides, many classification algorithms are very sensitive to the missing values. Because of these, handling the missing values is an important phase in many classification or data mining task. The experimental results prove a stable algorithm for improving the data classification accuracy on large datasets, which contain a lot of missing values.

Missing values exist in many situations wherein no values are reserved for some variable in an experiment or observation.

1. In real-life data, some stored values are frequently missing as a result of unexpected mistakes, most often because either they are lost or they are independent of conditions
2. Although missing values are a common occurrence, they can nonetheless have a significant effect on the processing of and results derived from data. First, the data mining program will constantly lose a considerable amount of useful information. Second, the system shows more signs regarding the uncertainty of the result, and it is difficult to ensure the determinateness.
3. Third, the missing values have a high probability of confusing the data mining process, leading to uncertain output. Fourth, the missing values frequently infect the operating performance and produce mistakes in the mining model.
4. Besides, some classification algorithms, such as back propagation neural network, K-Nearest neighbor algorithm, C4.5 decision tress and so on, are very sensitive to the missing values. If there are a lot of missing values in datasets, then use one of among algorithms to classify, let will have a high probability to obtain the low classification accuracy
5. Accordingly, handling missing values is an important step in preprocessing phases for most data classification or data mining tasks.
6. Inappropriate implementation of missing values can produce serious errors or false results.
7. Generally, methods for dealing with missing values can be divided into three classes:
   i) Delete the missing values;
   ii) Implement the missing values with estimated values; and
   iii) Ignore the missing values
8. Among these methods, deleting missing values is the easiest. However, when the rate of missing values in each attribute is high, this method has an unsatisfactory performance.
9. Ignoring missing values also causes similar issues. Thus naturally prefer methods for implementing missing values. There are many methods for accomplishing this, such as the approximation, stochastic regression, and neural network methods. Among all the approaches, the EM (expectation-maximization) algorithm can reliably use the stable and the maximum step to find the optimal values for implementing the missing values
10. However, the EM algorithm's speed of convergence is quite slow and easily falls into local optimization. Using EM algorithm fixed initial values, it can increase the speed of convergence and algorithm stability. At the same time, it can overcome the deviation by way of marginal values. Together, these give the EM algorithm a better performance. This improved EM algorithm is based on Naive Bayesian, and therefore is named the NB-EM algorithm, which uses the result of classification to substitute otherwise-random initial values. Below, it describe both the traditional EM algorithm and the NB-EM algorithm.

V.DATA IMPLEMENT AND CLASSIFICATION RESULT

In this experiment, select two datasets, both of which were downloaded from the UCI machine learning website. The first dataset describes kernels belonging to two different varieties of wheat: Kama and Rosa, 70 samples each and randomly selected. The second dataset describes vertebral columns divided into two categories: Normal (100 patients) and Abnormal (210 patients). Details about these two datasets are shown in Table 1 and Table 2.

Variable	Description
A	Different area
P	perimeter
C	Compactness C = 4*π*A/P^2
Length	Length of kernel
Width	Width of kernel
a	Asymmetry coefficient
l	Length of kernel groove

Table 1. Seed's Attribute Information

Variable	Description
PI	Pelvic incidence
PT	Pelvic tilt
LLA	Lumbar lordosis angle
SS	Sacral slope
PR	Pelvic radius
GS	Grade of spondylolisthesis

Table 2. Column's Attribute Information

For obvious results, let use MCAR method to increase the rate of missing values by upto 30%, and compared results of both traditional EM algorithm and our NB-EM algorithm to the datasets prior to values being removed. In applying to the MCAR Seeds' dataset both the traditional EM algorithm and the NB-EM algorithm, obtain two sets of optimal estimation shown in Table 3.

	Attributes	A	P	C
EM	Class A	10.8269840	10.7887897	0.58293985
	Class B	11.5229968	9.87305443	0.62747953
NB EM	Class A	10.64784974	9.329911901	0.608751284
	Class B	11.66546843	11.2472513	0.602245701
	Length	Width	a	l
EM	4.10955	2.73642423	2.548285411	3.786276617
	3.86550	2.782240497	1.93333857	3.176935235
NB EM	4.07118	3.331424247	2.74458494	3.432135692
	3.89497	2.138882163	1.689669352	3.530836093

Table 3. Seeds' Optimal Estimation with EM and NB-EM

In applying to the MCAR Columns' dataset both the traditional EM algorithm and the NB-EM algorithm, we obtain two sets of optimal estimation shown in Table 4.

	Attributes	PI	PT	LLA
EM	Class A	38.970776	11.66092049	33.61154508
	Class B	43.945822	12.97999265	46.15201578
NB EM	Class A	38.991969	11.52869673	33.30749357
	Class B	43.906938	13.18351406	46.60853731
	Attributes	SS	PR	GS
EM	Class A	26.203338	87.24249788	0.640833259
	Class B	34.795804	82.3805435	42.22666898
NB EM	Class A	26.179402	87.26586852	0.653738189
	Class B	34.822559	82.35016052	42.15643263

Table 4. Columns' Optimal Estimation with EM and NB-EM

Then use these two tables to substitute the missing values in MCAR datasets, obtaining two pairs of updated datasets. Subsequently, the expert input the updated datasets into Weka and classify them.

Implement the Missing Values

When the whole function achieves the convergence, there will be two outputs, first is the clustering result, second is the optimal values, which are used to replace the missing values. Firstly, expert should search from clustery table and obtain the sample belong to which class. Secondly, extract the corresponding attribute optimal values to replace the missing values in this sample. For example, the second attribute of this sample is missing, then, use the second optimal value to replace. After this phase, obtain a implemented dataset without missing values.

Classification Results

Table 5 and Table 6 show the results of different methods of implementing the missing values using the Multilayer Perceptron as the classifier in Weka. The accuracy rate shows the method that has a better effect.

Dataset	Correctly Classified Instances
Original Dataset	79.2857%
Dataset with EM algorithm	81.4286%
Dataset with NB-EM	88.5714%

Table 5. Classification Results of Seeds

Dataset	Correctly Classified Instances
Original Dataset	69.6774 %
Dataset with EM algorithm	73.2258 %
Dataset with NB-EM	78.0645 %

Table 6. Classification Results of Column

In both tables, the first row is the result of classifying the MCAR dataset without any processes, just a Multilayer Perceptron method to classify (Original Dataset). The second row is the result of using a traditional EM algorithm to substitute the missing values, then using a Multilayer Perceptron method to classify (Dataset with EM algorithm). The third row is the result of using an NB-EM algorithm to substitute the missing values, then using a Multilayer Perceptron method to classify (Dataset with NB-EM algorithm).

VI.CONCLUSION

Selection of missing values imputation method highly depends on given dataset, structure of attributes and missing data mechanism. Missing data mechanism is a key factor to decide if missing values can be imputed using some of described methods. Missing data mechanism can be considered as missing completely at random, missing at random or not missing at random. If missing data mechanism is considered as not missing at random, imputation cannot be done without knowledge of this mechanism. Unfortunately missing data mechanism is usually unknown. Some analytical methods have their own mechanism for dealing with missing data so missing values imputation methods should be used only if necessary. The NB-EM algorithm is used fixed the initial values to make sure the whole program avoid the local optimization and the influence of marginal values. Through repeating the Expectation Step and the Maximization Step wants to continuously approximate the optimal values. Therefore, the NB-EM algorithm can achieve a better result. The application of these results to data mining and knowledge discovery could not only help to improve the selection of a method for handling missing values during the data preprocessing phases of different data structures, but also produce a more reliable and efficient decision-making process given the uncertainties and incompleteness in presenting data collections.

VII.REFERENCES

[1]. Data Pre-processing & Mining Algorithm, Knowledge & Data Mining & Preprocessing, 3rd edition, Han & Kamber.

[2]. Agrawal, Rakesh and Ramakrishnan Srikant, "Fast Algorithms for Mining & Preprocessing Assosiation Rules", Proceedings of the 20th VLDB Conference, Santiago, Chile (1994).

[3]. Salleb, Ansaf and Christel Vrain, "An Application of Assosiation Knowledge Discovery and Data Mining (PKDD) 2000, LNAI 1910, pp. 613-618, Springer Verlag (2000).

[4]. Agrawal, R., and Psaila, G. 1995. Active Data Mining. In Proceedings on Knowledge Discovery and Data Mining.
[5]. Gustavo E. A. P. A. Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence, 2003.
[6]. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning, 1999.
[7]. L. Breiman, J. H. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA, 1984.
[8]. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society B, 1977.
[9]. Jamshidian, M., & Bentler, P. M. (1999). Using complete data routines for ML estimation of mean and covariance structures with missing data. Journal of Educational and Behavioral Statistics.
[10]. Yuan, K.-H., & Bentler, P. M. (2000). Three likelihood-basedmethods for mean and covariance structure analysis with nonnormal missing data. Sociological Methodology.
[11]. W. Vach, "Missing values: statistical theory and computational practice", Computational Statistics, Edited by P. Dirschedl and R. Ostermann, Heidelberg, **(1994)**.

