ITCI:An Information Theory Based Classification Algorithm for Incomplete Data

Yicheng Chen, Jianzhong Li, and Jizhou Luo

Department of Computer Science and Technology
Harbin Institute of Technology
chenyicheng20@163.com, {lijzh,luojizhou}@hit.edu.cn

Abstract. In the field of data mining, classification is an important aspect which has been studied widely. However, most of the existing studies assumed the data for classification is complete, while in practice, a lot of data with missing values exists. When dealing with these data, deleting the incomplete instances will result in a reduction of available information and filling in missing values may introduce skew and errors. To avoid the above problems, it is of great importance to study how to classify directly with incomplete data. In the paper, an information theory based classification algorithm, ITCI, is proposed. ITCI calculates the initial uncertainty of each class and attributes' contribution to decrease class uncertainty in the training stage and then, in the testing stage, an instance is assigned to the class whose uncertainty is minimum after all of the attributes are taken into consideration. Extended experiments proved the effectiveness and feasibility of the proposed method.

Keywords: information theory, incomplete data, classification.

1 Introduction

Classification algorithms, as an important aspect of data mining, has been widely studied and applied to many fields. However, previous studies often assumed the available data is complete, thus did not take the missing values into account, but in real life, however, incomplete data is ubiquitous[1], for example, in an industrial test, part of the data may be lost because of mechanical or electronic failure; in medical field, doctors may not get all the required data due to lack of equipment or patients' physical condition; in a social survey, some respondents may refuse to provide part of information; for the lack of permission, database query can not get all the data needed etc. Thus, it is of great theoretical and practical importance to study how to classify directly with incomplete data.

The missing(incomplete) data mechanism can be divided into the following three groups[2]: missing completely at random(MCAR), missing at random(MAR), not missing at random(NMAR). MCAR occurs when the missing of a variable is independent of itself and any other external influences; the missingness of MAR is independent of the missing variables but traceable from other variables; NMAR happens when patterns of missingness is non-random and depends on the missing variables, which is the most common situation in real life.

F. Li et al. (Eds.): WAIM 2014, LNCS 8485, pp. 167–179, 2014.

Currently, the methods to deal with incomplete data in classification falls mainly into three aspects. (1) Deleting the incomplete instances[3], this method is simple, but it will lose the useful information contained in incomplete data. (2) Using statistical and machine learning methods to fill in values most likely to be[4][5][6], such as filling manually, filling with mean or median, regression filling, KNN filling, filling based on neural networks etc. In general, filling manually will brings small bias, but it is not feasible given a large dataset with many missing values; filling by mean or median does not fully reflect the data variability and ignores the association between attributes; regression filling assumes a regression relationship exists among complete items and missing items, which is often incorrect in practice; KNN filling needs to define a reasonable similarity measure and has a relatively high computational complexity; filling based on neural networks requires designing appropriate network architecture for specific missing modes and it is too complex and cumbersome to apply. (3) Training and classifying with incomplete data directly, such as those methods based on EM[7], decision tree[8], fuzzy C-means[9], support vector machines[10], Bayesian networks[11] and the nearest neighbor[12]. Those EM-based methods require that the probability density function and missing attributes must be given, besides, they are often complex to train and converge slowly; the ID3-based approach treats the missing values as a special one different from known ones, which does not fit the real world well and it is difficult to get optimum due to the lack of a global search; fuzzy C-means and support vector machine based methods need assumptions of missing data's distribution, which is often not available in practice, thus the application is limited; Bayesian networks based methods require domain knowledge and dependencies among variables must be known, otherwise, complex network structure will be produced, what's more, the network nodes will increase exponentially with the growth of variables, which will result in high maintenance cost; as for nearest neighbor based method, when data's dimension is high, the sample space will still appears to be sparse even the dataset is large and applying the method directly will result in poor performance.

Among the methods to classify directly with incomplete data, [13] found that Naive Bayes methods are most insensitive to missingness, but they rely on apriori probability density to make classification inferences, which results in a low accurate. [14] proposed a method named RBC, which estimates incomplete data by intervals. In this method, missingness mechanism is not required to meet MAR assumption because all possibilities of the incomplete values are considered. Though it has better classification accuracy, the calculation is relatively complicated. [15] proposed the NCC2 method, which has higher classification accuracy, but it requires the missingness mechanism is declared and assumes each attribute contributes to classification independently, however, when the assumption is not met, classification accuracy decreases sharply. Other studies for classification with incomplete data include rough set based methods[16][17], such methods don't require any assumptions of missingness mechanism, but they are inefficient and have poor scalability.

In this paper, an information theory based classification algorithm, ITCI, is proposed for incomplete nominal data. The basic idea of the algorithm is as follows. At first, an initial uncertainty is calculated for each class, then an instance's attributes are inspected one by one to reduce class uncertainty. When all attributes are used, the instance is assigned to the class whose uncertainty is minimum. During the training stage, ITCI estimates the initial uncertainty with the help of the incomplete records, meanwhile, it calculates attributes' attribution to decrease uncertainty and for missing attributes it gets expected contribution. In the classification stage, expected contribution is used to estimate the decrease of uncertainty for missing attributes. With these measures, ITCI need not to estimate missing values explicitly, at the same time, it makes full use of the information contained in incomplete instances. Extended experiments show that the accuracy and stability of the proposed method are significantly higher than RBC and NCC2, and the time complexity is comparable low.

The rest of the paper is organized as follows: Section 2 gives the related concepts of information theory, their properties and application in classification. Section 3 gives an information theory based classification algorithm, ITC, for complete data. Section 4 extends the methods presented in section 3 to get an algorithm, ITCI, for incomplete data. Section 5 presents the results and analysis of experiments; Section 6 is the conclusion.

2 Basic Concepts and Problem Definition

2.1 Basic Concepts of Information Theory

Definition 1. (*self-information*) *The self-information of a random event is defined as the negative logarithm of the event's probability, namely, if the probability of event* x_i *is* $p(x_i)$*, then its self-information is defined as:*

$$I(x_i) = -logp(x_i) \tag{1}$$

Definition 2. (*conditional self-information*) *For any events* x_i *and* y_j *in a join set* XY*, the conditional self-information of* x_i *given* y_j *is defined as:*

$$I(x_i|y_j) = -logp(x_i|y_j) \tag{2}$$

Definition 3. (*mutual information*) *For sets* X *and* Y *of discrete random events, the information* x_i *acquired given* y_j *is called mutual information, which is defined as:*

$$I(x_i; y_j) = log\frac{p(x_i|y_j)}{p(x_i)} \tag{3}$$

We can get from formula (3) that $I(x_i; y_j) = log\frac{1}{p(x_i)} - \frac{1}{p(x_i|y_j)}$, and then get

$$I(x_i; y_j) = I(x_i) - I(x_i|y_j) \tag{4}$$

Formula(4) implies that mutual information equals the result of subtracting conditional self-information from self-information, or in another way, mutual

information is a measurement of decreased uncertainty, namely, mutual information equals the result of prior uncertainty $log\frac{1}{p(x_i)}$ subtracting remaining uncertainty $log\frac{1}{p(x_i|y_j)}$. Mutual information has the following properties:

(1) Reciprocity:

$$I(x_i; y_j) = I(y_j; x_i) \tag{5}$$

(2) When event x_i and event y_j are mutual independent, the mutual information is zero, namely, $I(x_i; y_j) = 0$.

(3) Mutual information may be positive or negative. When the value is positive, it means the appearance of event y_j will certainly contribute to the appearance of event x_i, on the contrary, it is disadvantageous.

(4) The mutual information between two events may not exceed the self-information of either one.

$$I(x_i; y_j) \leq I(x_i) \tag{6}$$

$$I(x_i; y_j) \leq I(y_j) \tag{7}$$

Definition 4. (*conditional mutual information*) *The conditional mutual information of x_i and y_j given z_k in join set XYZ is defined as:*

$$I(x_i; y_j|z_k) = log\frac{p(x_i|y_j z_k)}{p(x_i|z_k)} \tag{8}$$

The mutual information of x_i and $y_j z_k$ is defined as:

$$I(x_i; y_j z_k) = log\frac{p(x_i|y_j z_k)}{p(x_i)} \tag{9}$$

$$I(x_i; y_j z_k) = I(x_i; y_j) + I(x_i; z_k|y_j) \tag{10}$$

Formula(10) implies that given the appearance of a pair of events $y_j z_k$, the information x_i will get is $I(x_i; y_j z_k)$, which equals the information x_i get from the appearance of y_j, add the information x_i get from z_k when y_j is known.

The above four definitions are based on single event, similarly, they can be extended to event sets. We leave them out due to the limitation of the space.

2.2 Use Information Theory to Solve Classification Problems

In classification problems, an instance's feature can be represented by a n-dimensional vector $x = \{x_1, x_2, x_3, \ldots, x_n\}$. The classification task is to assign a label in label set $C = \{C_1, C_2, \ldots, C_K\}$ to each instance. Usually, the task includes two stages: classifier's training and testing. Considering the testing stage, for an instance x, we may assign any of the K labels to it and have some degree of uncertainty at the same time. The self-information of classes, $I(c_k)$, can be used to measure these initial uncertainty. Then attributes are taken into consideration one by one, meanwhile, the uncertainty of the classes will change with

the adding of attributes. When all attributes are considered, we can get the final uncertainty of each class, namely $I\left(c_k|x_1x_2,\ldots,x_n\right)$, then the instance x is assigned to the class whose uncertainty is minimum. As for the training stage, we estimate self-information and conditional mutual information with the help of training instances and they will be used as arguments of the final classifier.

3 ITC: Information Theory Based Classification for Complete Data

Assume the input space $\mathcal{X} \subseteq R^n$ is a n-dimensional vector set and the output space is class label set $\mathcal{Y} = \{c_1, c_2, \ldots, c_K\}$. For each instance, classification algorithms take $x \in \mathcal{X}$ as input and get $y \in \mathcal{Y}$ as output. The training set is $T = \left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(N)}, y^{(N)}\right)\right\}$, which has N instances. Let X be a random vector defined in input space $\mathcal{X}$, Y be a random variable defined in output space $\mathcal{Y}$. ITC builds a classifier by learning the self-information $I\left(c_k\right)$ of each class and the mutual information $I\left(c_k; x\right)\left(k = 1, 2, \ldots, K\right)$ between class c_k and feature vector x.

Considering the estimation of $I(c_k)$, we need to get the probability $P(c_k)$ of class c_k , which can be estimated by the following formula:

$$P\left(Y = c_k\right) = \frac{1}{N}\sum_{i=1}^{N} I\left(y^{(i)} = c_k\right) \tag{11}$$

For $I(c_k; x)$, when x's dimension is 1, we can get the value following formula(3) and $P(Y = c_k|X = x) = \sum_{i=1}^{N} I\left(y^{(i)} = c_k, x^{(i)} = x\right) / \sum_{i=1}^{N} I\left(x^{(i)} = x\right)$ is the probability estimation. But when the dimension continues to grow, the number of parameters will increase exponentially, which means it is not feasible to estimate all of them efficiently. Let the number of different values for attribute x_i is p_i, $i = 1, 2, \ldots, n$, the number of possible values of Y is K, then the total count of arguments is $K\prod_{i=1}^{n} p_i$. Therefore, we take some approximation measures to simplify the estimation described above.

Denote the mutual information between c_k and feature vector x as $I\left(c_k; x\right)$:

$$\begin{aligned} I\left(c_k; x\right) &= log\frac{p\left(c_k|x_1, x_2, \ldots, x_n\right)}{p\left(c_k\right)} \\ &= log[\frac{p(c_k|x_1, x_2, \ldots, x_n)}{p(c_k|x_1, x_2, \ldots, x_{n-1})} \times \frac{p(c_k|x_1, x_2, \ldots, x_{n-1})}{p(c_k)}] \\ &= I(c_k; x_n|x_1, x_2, \ldots, x_{n-1}) + I(c_k; x_1, x_2, \ldots, x_{n-1}) \end{aligned} \tag{12}$$

Formula(12) implies that when the feature vector x is known, the decreased uncertainty $I(c_k; x)$ equals the sum of $I(c_k; x_1, x_2, \ldots, x_{n-1})$, which measures the decreased uncertainty get from the former $n-1$ dimensions, and the conditional mutual information $I(c_k; x_n|x_1, x_2, \ldots, x_{n-1})$, which measures the decreased uncertainty get from x_n when the former $n-1$ dimensions are given. Using formula(12) recursively, we can get:

$$I(c_k;x) = I(c_k;x_1) + \sum_{i=2}^{n} I\left(c_k;x_i|x_1,x_2,\ldots,x_{i-1}\right) \tag{13}$$

From the definition formula $I\left(c_k;x_i|x_1,x_2,\ldots,x_{i-1}\right) = log\frac{p(c_k|x_1,x_2,\ldots,x_i)}{p(c_k|x_1,x_2,\ldots,x_{i-1})}$, we can see the arguments also increased exponentially, here, we simplify it as follows:

$$I\left(c_k;x_i|x_1,x_2,\ldots,x_{i-1}\right) \approx I\left(c_k;x_i|x_{i-1}\right) \tag{14}$$

Formula (14) implies that when the former $i-1$ dimensions are given, the decreased uncertainty we get from x_i is approximated by the decreased uncertainty we get from x_i when x_{i-1} is given.

Theoretically, if the mutual information is estimated according to formula(13), the results will be sole no matter in which order the attributes are considered, however, they will differ from each other if estimated following formula(14). In fact, because of $I\left(c_k;x_i|x_1,x_2,\ldots,x_{i-1}\right) \leq I\left(c_k;x_i|x_{i-1}\right)$, we should find an optimal order to make the expectation of $I(c_k;x)$ minimum, which will enable the bias as low as possible. Let $|x_1|$, $|x_2|$,...,$|x_n|$ be the number of different values of $x_1,x_2,\ldots,x_n$, among which we denote the maximum one as x_{max}, so the complexity of enumeration and estimation is $O(Kn!x_{max}^2 n)$ and it will be $O(NKn!x_{max}^2 n)$ if we estimate the expectation of $I(c_k;x)$ additionally. When the feature vector's dimension is high, it is not hard to see that the calculation is costly, or even impossible, thus we proposed a heuristic attribute order.

Definition 5. (*expected mutual information*) *The expected mutual information between* x_i (*whose value can take any one of* $x_{i1},x_{i2},\ldots,x_{ip}$) *and class* c_k *is defined as:*

$$E\left(c_k;x_i\right) = \sum_{r=1}^{p} p\left(x_i = x_{ir}|c_k\right) I\left(c_k;x_i = x_{ir}\right) \tag{15}$$

Definition 6. (χ^2 *of attribute pair*) *Assume* x_i *and* x_j *can take any value from* $x_{i1},x_{i2},\ldots,x_{ip}$, $x_{j1},x_{j2},\ldots,x_{jq}$ *respectively and let* n_{rs} *denote the number of instances in class* c_k *that satisfies* $x_i = x_{ir}$,$x_j = x_{js}$(r=1,2,... , p, s=1,2,... , q), *then we define the* χ^2 *value of attribute pair* x_i *and* x_j *as:*

$$\chi^2 = \sum_{r=1}^{p}\sum_{s=1}^{q} \frac{\left(n_{rs} - Np_{rs}\right)^2}{Np_{rs}} \tag{16}$$

In the formula above, p_{rs} denotes the expected joint probability of x_{ir} and x_{js} when they independent of each other, which can be estimated by $p_{rs} = \frac{1}{N^2}\sum_{k=1}^{p} n_{ks}\sum_{k=1}^{q} n_{rk}$.

For a set of attributes, $A = \{x_1,x_2,\ldots,x_n\}$, we give a heuristic algorithm to find the optimal order S for class c_k as follows:

ATT_ORDER select the attribute with maximum mutual information as the first one for it can decrease the uncertainty largely. In the following process, it selects the attribute which has the largest χ^2 value with the last selected one, in that this can make $I(c_k;x_i|x_{i-1})$ closer to $I(c_k;x_i|x_1,x_2,\ldots,x_{i-1})$ than other choices, so the total approximation error is small.

Algorithm 1. ATT_ORDER

Intput: $A = \{x_1, x_2, \ldots, x_n\}$ is the attribute set to be ordered
Output: optimal attribute order S

1. Calculate the expected mutual information of c_k and $x_i (i = 1, 2, \ldots, n)$, which is denoted as $E(c_k; x_i)$, then, choose the attribute x_j with the maximum value and add it to S, set the last selected attribute x_{last} as x_j.
2. For $k = 2$ to $n-1$, calculate the χ^2 value between x_{last} and each of the left attribute in A, choose the attribute x_j with maximum value and add it to S, set x_{last} as x_j.
3. Add the only attribute left in A to S.

On the basic of ATT_ORDER, we can get an optimal attribute order from the training data. Here, an algorithm named ITC is given for complete data as follows. The algorithm is made up of two parts, ITC_LEARN, which is used for learning model arguments, and ITC_TEST, which is used for applying the learnt model to classify instances with unknown labels.

Algorithm 2. ITC_LEARN

Intput: training data set $T = (x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$
Output: arguments $I(c_k), I(c_k; x_1), I(c_k; x_i|x_{i-1}) (k = 1, 2, \ldots, K, i = 2, 3, \ldots, n)$

1. Determine the optimal attribute order $S = x_1, x_2, \ldots, x_n$ by calling ATT_ORDER;
2. Calculate $I(c_k)$ using formula(1);
3. Calculate $I(c_k; x_1)$ using formula(3);
4. Calculate $I(c_k; x_i|x_{i-1})$ using formula(8);
5. Return all the calculated arguments.

4 ITCI: Information Theory Based Classification Algorithm for Incomplete Data

For incomplete data, we assume the missing mechanism to be MAR. The missing items can be any of the attributes of X or class label Y. In the same way, one or more attributes can be missing from feature vector X in the testing set.

We can get the algorithm, ITCI, based on ITC proposed in section 3. The main improvements include two parts: the estimation of statistic used to calculate model arguments; the estimation of decreased uncertainty of missing attribute. Once the estimations are acquired, ITCI can be trained and tested in the same way as ITC.

Algorithm 3. ITC_TEST

Intput: the feature vector x to be classified

Output: the predicted class label c of instance x

1. Calculate the initial uncertainty $I(c_1), I(c_2), \ldots, I(c_K)$ if x is classified to $c_1, c_2, \ldots, c_K$ without considering any attribute;
2. For each class, calculate $I(c_k; x_1), I(c_k; x_i|x_{i-1})(i = 2, 3, \ldots, n)$ according to the optimal attribute order $S = x_1, x_2, \ldots, x_n$;
3. Calculate the ultimate uncertainty by using formula: $I(c_k|x) = I(c_k)$ - $I(c_k; x)$ for each class. $I(c_k; x)$ can be estimated by $I(c_k; x) = I(c_k; x_1) + \sum_{i=2}^{n} I(c_k; x_i|x_{i-1})$;
4. Return the class whose $I(c_k|x)$ is minimum.

4.1 Estimations of Statistic

It can be seen from formulas (1)(3)and(8) that the key point of estimating model arguments, which include $I(c_k)$, $I(c_k; x_i)$, $I(c_k; x_j|x_i)$, is the estimation of frequencies if we use frequency to approximate probability. The main estimation includes $f(c_k)$, $f(c_k x_{ir})$, $f(c_k x_{ir} x_{js})$, $f(x_{ir})$, $f(x_{ir} x_{js})$, we denote the estimated values as $g(c_k)$, $g(c_k x_{ir})$, $g(c_k x_{ir} x_{js})$, $g(x_{ir})$, $g(x_{ir} x_{js})$ respectively.

Let the non-empty values of x_i to be x_{i1}, x_{i2},...,x_{ip} and the empty value to be x_{ip+1}, then denote the number of instances with value x_{i1}, x_{i2},...,x_{ip}, x_{ip+1} on x_i as $f(x_{i1})$, $f(x_{i2})$,..., $f(x_{ip})$, $f(x_{ip+1})$. Due to the existence of missing values, we intend to replace the first p frequency with $g(x_{i1})$, $g(x_{i2})$,...,$g(x_{ip})$, the estimation formula is as follows:

$$g(x_{ir}) = f(x_{ir}) + f(x_{ip+1}) \times f(x_{ir}) / \sum_{u=1}^{p} f(x_{iu}) \tag{17}$$

Let the non-empty values of x_j to be x_{j1}, x_{j2},..., x_{jq} and the empty value to be x_{jq+1}. Denote the frequency of instances whose value is x_{ir} on attribute x_i and is x_{js} on x_j as $f(x_{ir} x_{js})(r = 1,2,\ldots,p,\ s=1,2,\ldots,q)$. Then the estimation formula to assign in proportion is:

$$\begin{aligned} g(x_{ir} x_{js}) = {} & f(x_{ir} x_{js}) + f(x_{ir} x_{js}) \times f(x_{ir} x_{jq+1}) / \sum_{v=1}^{q} f(x_{ir} x_{jv}) + \\ & f(x_{ir} x_{js}) \times f(x_{ip+1} x_{js}) / \sum_{u=1}^{p} f(x_{iu} x_{js}) + \\ & f(x_{ir} x_{js}) \times f(x_{ip+1} x_{jq+1}) / \sum_{u=1}^{p} \sum_{v=1}^{q} f(x_{iu} x_{jv}) \end{aligned} \tag{18}$$

As for $f(c_k)$, $f(c_k x_{ir})$, $f(c_k x_{ir} x_{js})$, we estimate as follows. Assume G to be the set consist of T's instances whose class label is not missing and T_{c1},T_{c2},...,T_{cK} to be the sets get from partitioning G by class label. All the instances whose class label is missing are assigned to T_{cK+1}. Denote the number of instances in T_{ci} as $|T_{ci}|(i = 1,2,\ldots,\ K,\ K{+}1)$. Then the frequency estimation of c_k is:

$$g(c_k) = |T_{ck}| + |T_{cK+1}| \times |T_{ck}| / \sum_{i=1}^{K} |T_{ci}| \tag{19}$$

The estimation of $f(c_k x_{ir})$ is related to T_{ck} and T_{cK+1}. We treat all of the instances' class label of T_{cK+1} as c_k and assign a weight as follows:

$$w_k = |T_{ck}| / \sum_{i=1}^{K} |T_{ci}| \tag{20}$$

The frequency estimation $g_{ck}(c_k x_{ir})$ of T_{ck} can be estimated following formula (17), and the frequency of T_{cK+1} can be estimated as follows:

$$g_{cK+1}(c_{K+1} x_{ir}) = f_{cK+1}(x_{ir}) + f_{ck}(x_{ir}) \times f_{cK+1}(x_{ip+1}) / \sum_{u=1}^{p} f_{ck}(x_{iu}) \tag{21}$$

The subscripts c_k, c_{K+1} indicate that the frequency is estimated based on the dataset T_{ck}, T_{cK+1}. Formula (21) implies that we get the frequency of instances whose class label is missing based on the proportion of complete instances. Combining(20)(21), we can get the final estimation of $g(c_k x_{ir})$ like:

$$g(c_k x_{ir}) = g_{ck}(c_k x_{ir}) + w_k \cdot g_{cK+1}(c_{K+1} x_{ir}) \tag{22}$$

In the same way, the estimation of $f(c_k x_{ir} x_{js})$ also contains two parts, we can get $g_{ck}(c_k x_{ir} x_{js})$ following formula(18) from dataset T_{ck} and get the weight w_k following formula(20), and then get the estimation $g_{cK+1}(c_{K+1} x_{ir} x_{js})$ from dataset T_{cK+1} in a similar way as formula(18), but the weight is acquired from complete instances. Finally, $g(c_{K+1} x_{ir} x_{js})$ can be estimated like:

$$g(c_k x_{ir} x_{js}) = g_{ck}(c_k x_{ir} x_{js}) + w_k \cdot g_{cK+1}(c_{K+1} x_{ir} x_{js}) \tag{23}$$

4.2 The Arguments Estimation for Missing Attributes

Due to the existence of missing attribute, the arguments we need to estimate also include $I(c_k; x_{ip+1})$, $I(c_k; x_{jq+1}|x_{ir})$, $I(c_k; x_{js}|x_{ip+1})$, $I(c_k; x_{jq+1}|x_{ip+1})$. Here, we use the expected mutual information of all known attribute pairs as the estimations.

$$I(c_k; x_{ip+1}) = \sum_{u=1}^{p} p(x_{iu}|c_k) I(c_k; x_{iu}) \tag{24}$$

$$I(c_k; x_{jq+1}|x_{ir}) = \sum_{v=1}^{q} p(x_{jv}|c_k x_{ir}) \tag{25}$$

$$I(c_k; x_{js}|x_{ip+1}) = \sum_{u=1}^{p} p(x_{iu}|c_k x_{js}) I(c_k; x_{js}|x_{iu}) \tag{26}$$

$$I(c_k; x_{jq+1}|x_{ip+1}) = \sum_{u=1}^{q} \sum_{v=1}^{p} p(x_{iu} x_{jv}|c_k) I(c_k; x_{jv}|x_{iu}) \tag{27}$$

$I(c_k; x_{ir})$, $I(c_k; x_{js}|x_{ir})$ can be acquired based on the estimations given in section 4.1 and formula(1)(3)(8). The conditional probabilities $p(x_{ir}|c_k)$ can be estimated by the frequency of the instances whose value is x_{ir} on attribute x_i in class c_k. Other conditional probabilities can be estimated in a same way.

5 Experiment and Analysis

In order to evaluate the effectiveness of the algorithm, we did experiments on 12 datasets.The datasets are downloaded from the UC Irvine Machine Learning Repository and their basic information is given in table 1. Notice that all datasets are with missing values, among them, those whose names ended with 5 are acquired by randomly deleting 5% attributes from complete instances, others are incomplete initially. For continuous attributes, we discrete them in the preprocessing stage. All algorithms are implemented with java and run on a PC with 2.2GHZ cpu, 2GB memory and the operating system is Windows XP.

Table 1. The experimental datasets with missing values

dataset	instance number	class number	attribute number
breast_cancer	286	2	9
credit	690	2	15
cylinder	512	2	39
colic	368	2	22
mushroom_5	8124	2	22
wbdc_5	569	2	30
vote	435	2	16
crx_5	690	2	15
car_5	1728	4	6
nursery_5	12960	5	8
balance_5	625	3	4
vehicle_5	846	4	18

Table 2 shows the accuracy and standard deviation of the classifiers. In our experiments, the proposed algorithm ITCI and two classical classification algorithms dealing with incomplete data named RBC, NCC2 are compared. All the results are got from 10 times 10 fold cross-validation.

(1) We can see that the only dataset on which the accuracy of ITCI is lower than RBC is balance_5 and the difference is 2.90%. While on other 11 datasets, the accuracy is significantly higher, especially on vehicle_5 which exceeded more than 13.09%. Comparing ITCI and NCC2, we find that ITCI is lower on datasets credit and balance_5, while outstands significantly on the left 10. Especially on vehicle_5, it increased by 10.03%. For dataset balance_5, careful analysis found that the number of the three classes take proportions of 46.08%, 7.84%, 46.08% respectively. As the proportion of class 2 is too small, the initial uncertainty is relatively high, what's more, the number of attributes is too small to decrease

uncertainty during the classification process, so instances of class 2 are easily misclassified and then the final accuracy is affected. In fact, attributes of balance_5 are numerical and the class label is determined by the difference of the product of the first two attributes and the product of the last two attributes.While in ITCI, we assumed the attribute is nominal, so it is this inconsistency led to a low classification accuracy.

(2) By comparing the standard deviation of ITCI, RBC and NCC2,we find that ITCI is much better except on datasets nursery_5 and balance_5. By Analyzing dataset nursery_5, we find it also has the problem of imbalance classes, which leads to a large deviation of the initial uncertainty, thus the standard deviation is relatively high. But on the left 10 datasets, the standard deviation is small than or equal to the latter two.

Table 2. The comparison of classification accuracy and standard deviation

dataset	RBC	NCC2	ITCI
breast_cancer	72.70 ± 7.39	73.72 ± 7.71	78.12 ± 6.24
credit	86.49 ± 3.74	87.09 ± 3.80	86.57 ± 2.56
cylinder	76.09 ± 5.68	75.37 ± 10.28	81.49 ± 7.57
colic	79.59 ± 5.87	80.32 ± 5.69	83.25 ± 3.07
mushroom_5	95.65 ± 0.71	99.18 ± 0.32	99.65 ± 0.13
wbdc_5	95.38 ± 2.71	96.09 ± 2.50	96.41 ± 1.29
vote	90.16 ± 4.23	90.33 ± 4.14	94.42 ± 1.68
crx_5	85.22 ± 4.25	86.09 ± 4.23	86.68 ± 2.01
car_5	83.87 ± 2.04	85.38 ± 2.05	90.36 ± 1.63
nursery_5	87.75 ± 0.86	87.85 ± 0.85	88.59 ± 2.98
balance_5	89.62 ± 1.87	92.82 ± 2.56	86.72 ± 3.91
vehicle_5	62.83 ± 4.45	65.89 ± 4.57	75.92 ± 4.49

(3) By comparing the total time consumption(Details are not presented due to space limitation), we find NCC2 has the highest efficiency, RBC has the middle and ITCI has the lowest. But we also notice the total running time of ITCI for 10 times 10 fold cross-validation is 19.547s on nursery_5 which has 12960 instances. On average, an experiment takes only 0.195s, which implies the efficiency is still relatively high. In fact, ITCI can get all the arguments needed by scanning datasets only once, that means the complexity is not high at all.

Combining the above three comparison, we can draw the conclusion that the proposed algorithm, ITCI, is more accurate and stable than RBC and NCC2. Although the efficiency of ITCI is lower than the latter two, the running time and complexity is still relatively low, so it is useful in practice.

6 Conclusion

In the paper, an information theory based classification algorithm for incomplete data, ITCI, was proposed. ITCI treats classification as a process of decreasing

uncertainty, it calculates classes' initial uncertainty at first, then attributes are inspected one by one to decrease the uncertainty, and then an instance is assigned to the class whose uncertainty is minimum. In the training stage, ITCI weights frequencies by proportion, which makes full use of the information contained in incomplete instances. What's more, ITCI estimates the decreased uncertainty of missing attributes by expected mutual information. Experiments show that ITCI is more accurate and stable than existing ones and the time complexity is low, thus it is considered to be simple and practical. Our future work is to study classification with incomplete data for continuous attributes.

References

1. Gantayat, S.S., Misra, A., Panda, B.S.: A study of incomplete data – A review. In: Satapathy, S.C., Udgata, S.K., Biswal, B.N. (eds.) FICTA 2013. AISC, vol. 247, pp. 401–408. Springer, Heidelberg (2014)
2. Graham, J.W.: Missing Data Theory. Missing Data, pp. 3–46. Springer, New York (2012)
3. Little, R.J.A., Rubin, D.B.: Statistical analysis with missing data (2002)
4. Farhangfar, A., Kurgan, L.A., Pedrycz, W.: A novel framework for imputation of missing values in databases. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 37(5), 692–709 (2007)
5. Zhang, S., Jin, Z., Zhu, X.: Missing data imputation by utilizing information within incomplete instances. Journal of Systems and Software 84(3), 452–459 (2011)
6. Garca-Laencina, P.J., Sancho-Gmez, J.L., Figueiras-Vidal, A.R.: Pattern classification with missing data: a review. Neural Computing and Applications 19(2), 263–282 (2010)
7. Zhang, X., Song, S., Wu, C.: Robust Bayesian Classification with Incomplete Data. Cognitive Computation, 1–18 (2013)
8. Quinlan, J.R.: C4. 5: programs for machine learning. Morgan Kaufmann (1993)
9. Ichihashi, H., Honda, K., Notsu, A., et al.: Fuzzy c-means classifier with deterministic initialization and missing value imputation. In: IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007, pp. 214–221. IEEE (2007)
10. Chechik, G., Heitz, G., Elidan, G., et al.: Max-margin classification of incomplete data. In: Advances in Neural Information Processing Systems: Proceedings of the 2006 Conference, vol. 19, p. 233. The MIT Press (2007)
11. Wang, S.C., Yuan, S.M.: Research on Learning Bayesian Networks Structure with Missing Data. Journal of Software 7, 11 (2004)
12. Jonsson, P., Wohlin, C.: An evaluation of k-nearest neighbour imputation using likert data. In: Proceedings of the 10th International Symposium on Software Metrics, pp. 108–118. IEEE (2004)
13. Blomberg, L.C., Ruiz, D.D.A.: Evaluating the Influence of Missing Data on Classification Algorithms in Data Mining Applications. SBSI 2013: Simpiósio Brasileiro de Sistemas de Informacao (2013)
14. Ramoni, M., Sebastiani, P.: Robust bayes classifiers. Artificial Intelligence 125(1), 209–226 (2001)

15. Corani, G., Zaffalon, M.: Naive credal classifier 2: an extension of naive Bayes for delivering robust classifications. DMIN 8, 84–90 (2008)
16. Dai, J., Xu, Q., Wang, W.: A comparative study on strategies of rule induction for incomplete data based on rough set approach[J]. International Journal of Advancements in Computing Technology 3(3), 176–183 (2011)
17. Grzymala-Busse, J.W., Hippe, Z.S.: Mining Incomplete Data A Rough Set Approach. Emerging Paradigms in Machine Learning, pp. 49–74. Springer, Heidelberg (2013)