Identifying Appropriate Methodologies and Strategies for Vertical Mining with Incomplete Data

FARIS ALQADAH, ZHEN HU, LAWRENCE J. MAZLACK
Applied Computational Intelligence Laboratory
University of Cincinnati
USA
mazlack@uc.edu

Abstract - Many data mining methods are dependent on recognizing frequent patterns. Frequent patterns lead to the discovery of association rules, strong rules, sequential episodes, and multi-dimensional patterns. All can play a critical role in helping corporate and scientific institutions to understand and analyze their data. Patterns should be discovered in time and space efficient manner. Discovered patterns have authentic value when they accurately describe data trends; and, do not exclusively reflect noise or chance encounters. Vertical data mining algorithms key advantage is that they can outperform their horizontal counterparts in terms of both time and space efficiency. Little work has addressed how incomplete data influences vertical data mining. Consequently, the quality and utility of vertical mining algorithms results remains ambiguous as real data sets often contain incomplete data. This paper considers how to establish methodologies that deal with incomplete data in vertical mining; additionally, it seeks to develop strategies for determining the maximal utilization that can be mined from a dataset based on how much and what data is missing.

Key-Words: incomplete data, vertical, data mining, ignorability, efficiency, privacy preserving, data sensitivity, maximal utilization, methodologies, strategies

1 Overview and Objectives

Mining frequent patterns is one of the essentials in many data mining applications. Frequent patterns lead to the discovery of association rules, strong rules, sequential episodes, and multi-dimensional patterns. All of these applications play a critical role in allowing corporate and scientific institutions to further understand and analyze the data that they have gathered. In today's dynamic world it is essential for these patterns to be discovered in both a time and space efficient manner. The authentic value of these discovered patterns derives from the fact that they accurately describe trends in the data and do not simply reflect noise or chance encounters.

Vertical mining algorithms have been proposed that veer away from the traditional horizontal transactional database format. The key advantage of vertical mining algorithms is that they have been shown to outperform their horizontal counterparts in terms of both time and space efficiency. However, to the best of our knowledge no work has addressed the issue of how incomplete data influences the vertical data mining process. Therefore the quality and utility of the patterns and rules discovered via vertical mining algorithms remains ambiguous for real data sets that contain incomplete data. Therefore, the purpose of this work is to determine several different methodologies that deal with incomplete data in vertical mining. Furthermore we wish to develop strategies for determining the maximal utilization that can be mined from a dataset based on how much and what data is missing. Both vertical mining and incomplete data have been studied extensively separately, no comprehensive study combining both works is available.

The long term goal of our work is to efficiently mine incomplete data, and provide quality measures to the user on the results of the mining. This long term goal entails mining any form of data, be it transactional, observational, spatial etc. and any form of data mining. This paper's is focus is restricted to vertical mining techniques in stationary transactional data, We believe this short term goal will significantly contribute towards the long-term goal due to the fact that many data types and tech-

niques have their origins in stationary transactional data.

The central hypothesis of this work is that we can use statistical methods to determine an upper-bound on the quality of the patterns that can be discovered while using vertical mining techniques. This hypothesis has been formulated on the basis that most mining methodologies entail a user-defined minimum support and confidence value. Moreover, by exploiting the inherent structure of stationary transactional data along with the user-specified values we believe that we will be able to estimate the distribution of the data up to a certain degree of confidence. Once we approximate the distribution of the data we will be able to proceed in two different directions. First we will be able "reconstruct" to the missing values in the dataset in order to attain more accurate frequent patterns. Second we will be able to output several metrics about the quality of the patterns mined with the missing data. For example, one such metric could be the standard error associated with the distribution of the frequent patterns.

The development of such methods is essential to data mining in today's world for two main reasons:

- First, real life datasets always contain missing information, due to device errors & failures, human error & oversight, or simply a lack of data.
- Second, the issue of data privacy and confidentiality continues to grow by the day. In this environment fields in databases may be intentionally missing or slightly distorted, to preserve privacy and confidentiality.

Given this, scalable methods that maintain high quality results are needed more than ever.

This work will be tested and implemented in the following ways:

1.1 Build statistical models that will enable us to approximate the distribution of the data.

Based on our preliminary knowledge and data, the working hypothesis is that computing sample statistical moments will enable us to approximate the distribution and character of the data up to a certain degree of confidence. This hypothesis is supported by the fact that we can model any categorical data set via a multivariate, multinomial probability distribution. Therefore by computing sample statistical moments such as the average and central moments such as variance we will be able to approximate the data distribution. Furthermore these calculations can be performed very efficiently and simultaneously while the vertical algorithm is working.

1.2 Use the statistical models to maximize the quality of vertical mining algorithms.

Building upon the previous working hypothesis we believe that we can maximize the quality of vertical mining on data in three ways:

- Output a quality measure of the mining performed
- If our approximate model has a confidence level we can actually "fill in" in the missing data fields without degrading the results.
- Report on the ignorability of the data.

This work is inventive and novel in its approach because it combines two previously unrelated fields and utilizes both to address real-life issues. The broad applications and positive impact of this work range from preservation of privacy and confidentiality to stronger correlation discovery in scientific and research databases. As computing and communication technology continue to grow, the amount of data collected also continues to grow at an exponential rate. Through the development of our technique scalable and accurate algorithms will evolve to maximize our utility from this data.

2 Background

2.1 Incomplete Data

Incomplete data handling has been a topic of significant interest for many years. Previous studies can be divided into two approaches. The first approach identifies many types of incomplete data and utilizes different methods to handle them. For example in [3], Grzymala-Busse specified that incomplete data is split into two broad categories, "missing" and "do not care". Furthermore he obtained characteristic block by eliminating tuples containing the incomplete data that is characterized as "missing," or replicating the tuples in every characteristic block, if it is characterized as

"do not care". However Grzymala-Busse's strategy looses the broad view of the database because they only consider situations of a single data tuple while not considering the aggregate effect of incomplete data. For example incomplete data happens that appears continuously in modern databases for confidentiality reasons are not considered in the work by Grzymala-Busse.

The other approach is to leave incomplete data as incomplete and use different statistical methods in order to achieve reasonable evaluation metrics. For example in [7], Quinlan use a literal to denote the incomplete data and a summary function after constructing a decision tree; furthermore he assigns values to the literals in order to achieve maximum correctness of decision on the training data. However, the question that is still left, is: What if the number of tuples with incomplete data is extremely large? In this case, this method seems infeasible to address the issue. Even if the number of incomplete data is small, it is very important to decide which rules in the decision tree are legal and which ones are heavily influenced by missing data. Unfortunately, previous work has failed to significantly quantify the degree to which incomplete data affects the entire mining process.

The question of ignorability in categorical data has been studied by [17, 18, 19, 20, 21] from a statistical and mathematical point of view. The key condition that they identified is that the data should be "missing at random" and "coarsened at random". In [18] Rubin also requires a parameter of distinctness as a second condition in evaluating ignorability. The work in ignorability in categorical data presents very interesting results; however most of the results depend on some pretty strong assumptions about the nature of the missing data. For example in [17] the authors present the result that ignorability can occur for maximum likelihood inference in categorical data, if the following assumptions are held: 1) the data is weakly coarsened at random or 2) the data is strongly coarsened at random. Furthermore the ignorability of the data depends on the inference method. For example in [18] ignorability is computed in terms of maximum-likelihood that plays an important role in the EM-algorithm. Nonetheless, we construct ignorability and confidence measures independent of the specific mining algorithm and free of assumptions about the missing data values.

2.2 Vertical Mining

As mentioned earlier, mining frequent patterns in datasets is a primary problem in data-mining applications. Traditionally the suggested pattern-mining algorithms have been variants of the Apriori [12] algorithm. Apriori uses a bread-first, bottom-up search that enumerates every single frequent itemset. The algorithm first enumerates all 1-itemsets, and then builds the set of 2-itemsets and iterating until to computes all frequent itemsets. Furthermore, the downward closure property of itemset support is used to prune candidates at each stage of the algorithm resulting in good performance by significantly reducing the number of candidates generated.

While the development of Apriori-inspired algorithms was sufficient for sparse datasets, they have not scaled as well when the datasets tend to be dense. The main problem with these methodologies is that in order to compute support of any itemset the program must constantly refer back to the datasets, incurring a high I/O cost. Secondly if the patterns of data are long, then it is computationally expensive to check large sets of candidates by pattern matching.

Most previous work on mining frequent patterns has utilized the conventional horizontal transactional database format. A number of vertical mining algorithms have been proposed recently for computing frequent patterns [10, 13, 14, 15, 16] using a vertical database format. In a vertical database each item is associated with it's corresponding id-set. Mining algorithms utilizing the vertical format have been shown to outperform algorithms using the horizontal format. The main advantage of the vertical algorithms stems from the fact that calculating the support of items can be performed efficiently using id-set intersections. Horizontal approaches, on the other hand, maintain complex data-structures such as hash and search trees. Furthermore, it was shown in [13] that for databases with long transactions, the vertical approach reduces the number of I/O operations. User-id sets or tidsets offer natural pruning of irrelevant transactions as a result set intersec-

tions. Also in [16] it was shown that using compressed vertical bitmaps for association mining outperforms in some cases even an optimal horizontal algorithm that had complete *a priori* knowledge of all frequent items and only needed to find their frequency.

Zaki [10] suggests a superior method of performing vertical mining. Instead of maintaining user-id sets, difference sets are utilized. These diffsets only keep track of the differences between user-id sets. It was shown experimentally by Zaki that diffsets cut down the size memory required to store intermediate results by orders of magnitude. The initial database is stored as a diffset format, which can easily fit into main memory. Furthermore since the diffsets are much smaller than regular user-id sets, intersection operators are performed extremely fast. It was additionally shown in [10] that with the use of the diffsets several vertical mining algorithms efficiency was increased by several orders of magnitude.

3 Importance

3.1 Problem Setting

Frequent pattern mining proceeds as the following. Let I be a set of items database where each transaction contains a unique id. A set $X \subseteq I$ and tidset (transaction identification set). An itemset with k items is referred to as a k-itemset. The support as $\sigma(X)$, is the number of transactions where it occurs as a subset; support is greater or equal to a user-specified *minimum support*

Frequent patterns are used to discover *association rules.* An association rule is an expression $X \rightarrow Y$ where X and Y are itemsets. Each rule also has associated with a support value s and a confidence value c. The support of a rule is the joint probability of a transaction containing both X and Y and is given as $s = s(XY)$. The confidence c of a rule is the conditional probability that a transaction contains Y, given that it contains X; and is given as

$$c = \frac{s(xy)}{s(y)} \quad (1)$$

Rules are labeled as frequent if their support is greater than a specified minimum support, and labeled as strong of their confidence is greater than a specified minimum confidence.

Association rule mining involves computing all the frequent patterns that are found in the dataset. When searching for interesting rules from large quantity of data, if the number of missing data points exceeds some threshold, then our mining result will be visibly inaccurate.

A possible way that incomplete data may influence the result, even though the number of missing data is small, is where the incomplete tuples are of greater significance than other "noisy" tuples. For example, as when association rules are mined, a discovered association rule must satisfy the requirements of minimum support and confidence. If the missing data is just on the boundary as to whether the association rule satisfies the minimum requirements, then we may classify this as a "significant" portion of missing data. In this case we can not just simply ignore the missing data.

We are concerned about following specific problems that may occur while trying to discover association rules from a dataset that contains incomplete data, and try to seek reliable solution to these problems:

3.1.1 If the incomplete data (or value) is a distinct value which never emerges in another data tuple, how we handle that?

If there is a distinct value missing in the database, then a vertical column in the vertical mining process will be lost, which will greatly influence the mining result.

Based on our observation, if the number of incomplete data is smaller than the user specified minimum support or confidence, we can just ignore this question and treat the incomplete as non-distinct data. But what if the number is so large that we cannot ignore them? What we will do is try to use a quasi number or use a literal denotes missing value. And after traversing the database, get a function with the literal or quasi number. Then we can use statistic method to check whether our assumption that there is large enough number of incomplete data has distinct value that do not appears before.

Using our statistical techniques we may draw assumptions about whether or not the incomplete data is or is not distinct.

3.1.2 How do we handle incomplete data, if such the *incompleteness comes from missing data*?

Many reasons will cause the data missing, such as disk damage, careless recording, and so on.

We will try two approaches in to solve the problem. And, subsequently use the method with best performance.

- First, use the most common data in the same vertical column and use substitute it for the missing data. This method is ad hoc and imperfect, but when the amount of missing data is small or medium; it may work well enough. The results should be verified through experiments.
- Second, use a literal substitute for the missing data, which will join every association rule or causal rule mining. For each rule, we will develop a minimum support or confidence with the literal. Through analysis of the function containing the literal, we can conclude how can we cope with the missing data: ignore, or assign with a specific value.

3.1.3 How do we handle incomplete data, if such the *incompleteness comes from "do not care" values that* originally do not need to be captured?

Many ways could also cause the "do not care" incomplete data, for example during data integration from multiple sources of data, one database might contain an attribute while the other does not contain.

3.1.4 How do we handle incomplete data, if the *incompleteness comes from confidential requirements*?

These situations may be a little difficult to understand, but consider the following example situation: Suppose that an insurance company and a bank wants to integrate some data and do data mining on the integrated data. They are separate business entities. The bank does not want to show the insurance company the exact account balance for each customer, while the insurance company wants to mining interesting rules based on financial assets and gender. How might this be done?

In this work, we treat data in this form as incomplete data. Using the example above, we could require that the bank provide the salary after adding a perturbation value for each customer, while ensuring that the summation of those values for each gender is zero. In this way, although the bank provides individual values, they do not show precisely accurate ones. However, the data mining question may still be asked. For example, how to handle the problem if the lift the attribute concept is as in [11]? The only possible solution we can foresee now is using granule theory; stratifying the data, and using the stratified data to do data mining.

3.1.5 How to assign a *threshold for the amount of missing data* that determines when we can ignore the missing data?

The incomplete data from various sources sometimes could influence our data mining algorithm, sometimes not.

We run the data mining algorithm first and at the same time count the number of missing data items. Compare rules, which satisfy the minimum metric (support & confidence) requirement by adding the incomplete data with the rules without incomplete data. If there are any differences, we can conclude the incomplete will influence our mining conclusion. Then we use different methods mentioned above to handle that. And then obtain new conclusions after filling in each incomplete data. Lastly, make a comparison between rules after filling with rules without filling. Try to find minimum number of incomplete to be filled which will be the threshold.

3.1.6 How to cope with the problems that there are so many *missing data appears continuously*?

It could possible when some data digestion equipment is out of order, which makes incomplete data happen continuously.

If the there are many incomplete data for one attribute, there are many assumptions we can make, among them:

- First, the missing data distribution is the same as what have already appeared. In this case, when we count minimum support or confidence, rather than filling a data for each incomplete data item, we can simply add the number times the distribution probability and skip all of the continuous incomplete data.

- Another possible assumption is that the data missed is the same as data pattern appeared just before or after. Then, we can use an appropriate method to fill the missing data and subsequently perform a mining algorithm.

Both of these assumptions can only be used after an experimental test, or treat them as a use specified parameter, which makes the user decide how to fill the incomplete data.

3.1.7 How to cope with the problems that, while the number of data items with incomplete value is small, it is important to vertical mining.

For example, suppose that after performing the data mining algorithm, we discovered that a pair of associated items could not be treated as an association rule because it was below the amount of specific data needed to meet minimum support threshold by only one item. While, at same time, there is one item set with incomplete data. In this case the missing data might be very important.

We assign a weight for each incomplete data item, especially when the number of incomplete data is small. Situations such as what we described above should be given higher weight. After assigning weights, we use one literal to denote incomplete data and literal multiply weights as values. And, then perform the mining algorithm; achieve functions of support and confidence with the variables of weight and value. Lastly, using mathematical or statistical tools, analyze the function; find the most optimized variable value that makes the function achieve optimized value.

3.2 Approach Significance

Strategies for handling incomplete data have been studied for a long time. However, combining the principles of vertical mining with incomplete data handling has not drawn attention before. Research into incomplete data has only focused on how to give reasonable values to substitute the incomplete data. However, what we are suggesting will substitute the incomplete data within the framework of any already existing vertical algorithm. Moreover, our suggested ideas allow for the development of metrics about the quality of results outputted based on real statistics located within the data. In this manner end users will know the value of the patterns they have extracted without having to refer to domain experts or running expensive real world quality tests. Currently no other work appears to accomplish either of these tasks.

This work is significant because it tackles two major problems facing the data mining community: scalability and efficient quantification of the true value of patterns extracted. This is an important problem because the ultimate goal of data mining is to infer sound, accurate, and previously unknown patterns in the data in an efficient manner. While vertical mining has produced efficient methodologies for mining, and incomplete data theory has provided a preliminary model on how to deal with data uncertainty, the end user has yet to see the combined effect.

The technical challenge of this problem is rooted in two main places:

- Traditionally building a statistical model of a dataset is an expensive proposition as far as computation is concerned. However we have the advantage that we do not have to construct this model explicitly from scratch. We utilize user specified minimum support and confidence values that greatly aid in our model construction. Furthermore, our model only needs to be sophisticated enough to attempt to recover the *missing* data tuples to reach a specified degree of confidence.
- The second challenge is to maintain the integrity of the vertical mining algorithm that we will run alongside our model computation. Our method will maintain and may bolster the results of any vertical mining algorithm, as we use the results of the algorithm to help construct our model.

Beyond the traditional notion of missing data, our work also entails incomplete data that appears continuously and consistently throughout datasets due to privacy concerns. Data mining activities today can involve several parties exchanging information in order to extract patterns useful to all parties. For example, consider a hospital and health insurance company that wish to exchange information about a common set of patients/clients. It is advantageous for both parties to trade information; however the privacy of the patients must be preserved. In order to maintain con-

fidentiality and privacy data can be intentionally left missing or distorted in the data that the hospital and insurance company swap. It is essential to quantify how much data and which data points can be intentionally left out and still maintain scalable, high quality mining results. Through our suggested framework, such quantification will be possible.

4 Long Term Goals

The long-term goals are to setup and design a complete strategy for handling incomplete data in various data formats and mining tasks. Through this work, we will provide insight into methodologies that can be used for any association rule mining process and any data type. This belief is based upon two main foundations. The ability to build an accurate statistical model of data is data-type independent. Our hypothesis is that we will be able to build statistical models to estimate the distribution of the transactional data is not dependent upon the type of data; rather it depends on calculating statistical moments and central moments such as the average, variance and covariance. Furthermore, all association rule mining processes require the user input of minimum support and minimum confidence. Using these inputs along with the statistical models that we build, we may infer the quality of the results mined. The difference between the data types will come into play in the methods used to calculate the support and confidence of the data. However the key components of an overall package for dealing with missing data in association mining we believe can be derived from the completion of this work. The following lists long term goals that our work contributes to:

- Provide recommended strategies of coping with the incomplete data in all possible environments.
- Inform users of several different statistical strategies that can be used to handle incomplete data based on the specific requirements of users.
- Compare different strategies on different types of data.
- Inform users and ask for decisions on how to handle incomplete data when the size of incomplete data is small but largely influences the result.
- Evaluation metrics for estimating the performance of the strategy handling the incomplete data.

5 Experimental Plans

5.1 Experiment for first objective: *statistical models that will enable data distribution approximation*

Our first specific objective was to *build statistical models that will enable us to approximate the distribution of the data.* We may view our transactional data as a multivariate multinomial experiment with the following properties:

- Each experiment consists of *n* identical trials
- Each trial results in one *k* (*k=2)* outcomes.
- The probability that a single trial will result in outcome *i* is π_i.
- The trials are independent.
- We are interested in n_i, the number of trials resulting in outcome *i*.

Given that our transactional database follows a multivariate multinomial distribution we can specify the number of observations resulting in each of the *k* outcomes by the following formula:

$$P(n_1, n_2, ...n_k) = \frac{n!}{n!n_1!...n_k!}\pi_1^{n_1}, x_2^{n_2}...\pi_k^{n_k} \qquad (2)$$

We refer to the π_i probabilities as *cell probabilities,* where one cell corresponds to each of the *k* outcomes. The observed numbers n_k will be called *observed cell counts*. By the properties of the multinomial distribution, the expected number outcomes of type *i* in *n* trials is $n\pi_i$. Utilizing all of these properties we may test our method of building a valid statistical model via the following experiment:

- Obtain a complete synthetic transactional dataset *S* and compute its complete multivariate multinomial probability distribution.
- Randomly "poke out" data values to simulate incomplete data in a real life situation, label this as *S'*.

- Utilizing sample moments such as average and variance compute hypothesized cell probabilities.
- We can now calculate the expected cell counts by using the properties of multinomial distribution as mentioned above.
- If the hypothesized π_i-values are correct the observed cell counts should deviate greatly from the expected cell counts. The chi-square goodness of fit test statistic measures this deviation and is:

$$\chi^2 = \sum_i \left[\frac{(n - E_i)^2}{E_i} \right] \quad (3)$$

where E_i is the expected cell count. The distribution of the quantity χ^2 can be approximated by a chi-square distribution, provided that the cell count is large. This is a reasonable assumption since we will be dealing with large transactional data sets. Using this property we may obtain upper-tail values and confidence intervals of how well our model fit the actual data with *k-1* degrees of freedom. These confidence intervals will indicate to us how well our hypothesized cell probability values match the actual values.

This experiment will be run several times with several different synthetic datasets and several different random distributions used to "poke out" data and simulate incomplete data. Through all of these experiments we will keep track of all the χ^2 quantities.

5.2 Experiment for second objective: *maximize the quality of vertical mining algorithms*

Our second specific objective is to use the developed statistical model from part 1 to *maximize the quality of vertical mining algorithms*. Specifically, we would like to produce a quality measure of the frequent patterns discovered with the missing data set. Also we wish to fill in the data with our hypothesized cell probabilities from part 1. The quality measure to be outputted will be exactly equal to χ^2 quantity developed earlier. Frequent patterns that were discovered that greatly exceeded the minimum support and minimum confidence levels would obtain a full score ranking, since any missing data would not affect these results. However frequent patterns that have support and confidence that is "close" to the minimum values will be ranked according to the χ^2 quantity computed over their respectful cell probabilities.

The following experiment will test the quality of the patterns that we mine using our computed statistical model:

- Obtain a complete synthetic transactional dataset *S* and compute its complete multivariate multinomial probability distribution, while running a vertical mining algorithm that will output frequent patterns. Label the result of the vertical mining algorithm *R*.
- Randomly "poke out" data values to simulate incomplete data in a real life situation, label this as *S′*.
- Re-run vertical mining algorithm on *S′*, and develop results *R′*.
- Compute hypothesized cell probabilities according to our statistical model.
- Calculate expected cell counts and fill in incomplete data, call this dataset *S″*.
- Re-run vertical mining algorithm on *S″*, and develop results *R″*.
- Compare *R* to *R′* to *R″* and measure the degree to which they match.

5.3 Future Directions

Our work revolves around the χ^2 statistic. Future work may attempt to define new statistics that are specifically tailored to association rules, and can integrate the user specified confidence and support values more tightly. Furthermore, the math and science behind log-linear analysis has been linked to multinomial distributions of categorical data for a long time. These models need to be investigated further in the context of association rules and frequent patterns, and not just in the context of ignorability of data. Our experiments are a starting point for any future work in this particular complex field of data mining.

6 Epilogue

Data mining technologies are currently being used in commercial, industrial, and governmental businesses for purposes, ranging from increasing profitability to enhancing national security. The widespread applications of data mining technolo-

gies has raised concerns about trade secrecy of corporations and privacy of innocent people contained in the datasets collected and used for the data mining purpose. It is necessary that data mining technologies designed for knowledge discovery across corporations and for security purpose towards general population has sufficient privacy awareness to protect the corporate trade secrecy and individual private information. Unfortunately, most standard data mining algorithms are not very efficient in terms of privacy protection, as they were originally developed mainly for commercial applications, in which different organizations collect and own their private databases, and mine their private databases for specific commercial purposes. The current methods utilized to preserve confidentiality and privacy while performing data mining revolves around creating missing data, in such a manner that privacy is preserved. The general goal privacy-preserving data mining techniques is defined as to hide sensitive individual data values from the outside world or from unauthorized persons, and simultaneously preserve the underlying data patterns and semantics so that a valid and efficient decision model based on the distorted data can be constructed.

Our work will accomplish both of these tasks. Our statistical model will be able to reproduce that data that was "poked out" in order to obtain significant results; while on the other hand the exact data values will never be known (since they are generated by our model), this will preserve privacy of users. The general goal privacy-preserving data mining techniques is defined as to hide sensitive individual data values from the outside world or from unauthorized persons, and simultaneously preserve the underlying data patterns and semantics so that a valid and efficient decision model based on the distorted data can be constructed.

7 References

[1] J. Stefanowski, A. Tsoukias [2001] "Incomplete information tables and rough classification," Computational Intelligence, vol. 17, no. 3, 545-566

[2] J. W. Grzymala-Busse [2003] "Rough Set Strategies to Data with Missing Attribute Values," *Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining,* Melbourne, FL, USA, 56–63,

[3] J. W. Grzymala-Busse, S. Siddhaye [2004] "Rough Set Approaches to Rule Induction from Incomplete Data," *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2004),* Perugia, Italy, vol. 2, 923–930

[4] M. Kryszkiewicz [1995] "Rough set approach to incomplete information systems," *Proceedings of the Second Annual Joint Conference on Information Sciences, Wrightsville Beach, NC*, 194–197

[5] M. Kryszkiewicz [1999] "Rules in incomplete information systems," *Information Sciences 113*, 271–292

[6] C.H. Cai, Ada W.C. Fu, C.H. Cheng and W.W. Kwong [1998] "Mining association rules with weighted items," *Proceedings of International Database Engineering and Applications Symposium (IDEAS 98)*, 68-77

[7] J.R, Quinlan [2003] "Induction of Decision Trees," *Machine Learning***,** vol. 1, issue 1, 81–106

[8] R. Agrawal, R.Srikant [2000] "Privacy-preserving data mining," In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data***,** *Dallas, Texas, United States,* 439-450, Dallas, Texas, United States

[9] R. Agrawal, T. Imielinski, A. Swami [1993] "Mining association rules between sets of items in large database," *Proceedings of the 1993 ACM SIGMOD international conference on Management of data***,** Washington, D.C., United States, 207-216, 1993

[10] M. J. Zaki, K. Gouda [2003] "Fast Vertical Mining Using Diffsets," In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, D.C, 326-335

[11] J. Han, Y. Cai, N. Cercone, "Knowledge Discovery in Databases: An Attribute-Oriented Approach," *Proceedings 18th Int. Conference Very Large Data Bases*, 547-559, 1992

[12] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. Inkeri Verkamo [1996] "Fast discovery of association rules" In U. Fayyad and et al (editors), *Advances in Knowledge Discovery and Data Mining*, 307-328. AAAI Press, Menlo Park, CA

[13] B. Dunkel, N. Soparkar [1999] "Data organization and access for efficient data mining" *15th IEEE Intl. Conf. on Data Engineering*, March

[14] P. Shenoy, J.R. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa, D. Shah [2000] "Turbo-charging vertical mining of large databases," *Proceedings ACM SIGMOD Intl. Conf. Management of Data*, Dallas

[15] M.J. Zaki [2000] "Scalable Algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12 (3): 372-390, May-June

[16] M.J. Zaki, C.J. Hsiao [1999] CHARM: An efficient algorithm for closed association rule mining," Technical Report 99-10, Computer Science Dept. Rensselaer Polytechnic Institute, October

[17] M. Jaeger [2005] "Ignorability for Categorical Data," *The annals of Statistics*, Vol. 33 No.4

[18] D. Rubin [1962] "Inference and missing data (with discussion)," *Biometrika* 63 581-592

[19] D.F. Hetijan, D.B. Rubin [1991] "Ignorability and coarse data," *Annals of Statistics* 19 2244-2253

[20] D.F. Heitjan [1994] "Ignorability in general incomplete-data models," *Biometrika* 81 701-708

[21] D.F. Heitjan [1997] "Ignorability, sufficiency and ancillarity," *J. Roy Statistical Society Ser.* B 59 375-381

[22] M. Houtsma, A. Swami [1995] "Set-Oriented Mining of Association Rules in Relational Databases," *11th Int'l Conf. Data Eng*

[23] S. Sarawagi, S. Thomas, R. Agrawal [1998] "Integrating Association Rule Mining with Databases: Alternatives and Implications" *ACM SIGMOD Int'l Conference Management of Data*, June

[24] M.J. Zaki, S. Parthasarrathy, W. Li, M. Ogihara [1997] "Evaluation of Sampling for Data Mining of Association Rules" *Seventh Int'l Workshop on Research Issues in Data Eng.*, April

[25] V. Ganit, J. Gehrke, R. Ramakrishnan [1999] "CACTUS: clustering categorical data using summaries" *SIGKDD Conf.*