COMBINED MINING APPROACH TO GENERATE PATTERNS FOR COMPLEX DATA

Sumit Kumar*, Sweety×, Manish Kumar×

IVY Comptech Pvt. Ltd.-Hyderabad*, IIIT-Allahabad×, India
[sumit78kumar90, sweety.bhagat89, manish.singh76]@gmail.com

ABSTRACT

In Data mining applications, which often involve complex data like multiple heterogeneous data sources, user preferences, decision-making actions and business impacts etc., the complete useful information cannot be obtained by using single data mining method in the form of informative patterns as that would consume more time and space, if and only if it is possible to join large relevant data sources for discovering patterns consisting of various aspects of useful information. We consider combined mining as an approach for mining informative patterns from multiple data-sources or multiple-features or by multiple-methods as per the requirements. In combined mining approach, we applied Lossy-counting algorithm on each data-source to get the frequent data item-sets and then get the combined association rules. In multi-feature combined mining approach, we obtained pair patterns and cluster patterns and then generate incremental pair patterns and incremental cluster patterns, which cannot be directly generated by the existing methods. In multi-method combined mining approach, we combine FP-growth and Bayesian Belief Network to make a classifier to get more informative knowledge.

KEYWORDS

Association Rule Mining, Lossy-Counting Algorithm, Incremental Pair-Patterns, Incremental Cluster-Patterns, FP-growth, Bayesian Belief Network.

1. INTRODUCTION

In real-time data mining algorithms, data sampling generally not accepted because then it may miss some important data that may be filtered out during sampling. If we have to deal with some distinguish large data sets then joining of those data sets into one data set may not be possible as that would be more time and space consuming. More often this approach of handling multiple data sources can only be developed for specific cases and cannot be applied for all problems. Combined mining is a two-to-multistep data mining approach, which involves first mining the atomic patterns from each individual data source and then combines those atomic patterns into combined-patterns by pattern-merging method, which is more suitable for a particular problem. In multi-source combined mining approach, we first find the informative patterns from individual data source and then generate the combined patterns, which can't be directly generated by some traditional algorithms like FP-growth etc. In multi-feature combined mining approach, we consider features from multiple data sets while generating the informative patterns, where it is necessary in order to make the patterns more actionable. In case of cluster patterns, we made the cluster of patterns with same prefix but the remaining data items in the pattern make the results to be different. The main advantage of our approach is that we don't need to apply neither any

Jan Zizka (Eds) : CCSIT, SIPP, AISC, PDCTA - 2013
pp. 91–97, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3609

pruning method nor any clustering method separately to get the more informative patterns as during the lossy-counting algorithm's implementation, we have already prune at the boundary of the data sources and we also get the most similar data items in the same bucket itself.

2. RELATED WORKS

Kargupta and Park (2002) provide an overview of distributed data mining algorithms, systems and applications. They pointed out a mismatch between the architecture of most off-the-shelf data mining systems and the needs of mining systems for distributed applications. They also claim that such a mismatch may cause a fundamental bottleneck in many distributed applications. Kargupta et al (1999) presented a framework of collective data mining to conduct distributed data mining from heterogeneous sites. They point out that in a heterogeneous environment, naïve approaches to distributed data analysis may lead to incorrect data-model. Chattratichat et al (1999) designed Kensington software architecture for distributed enterprise data mining, which addresses the problem of data mining on logical and physical distribution of data and heterogeneous computational resources. Karypis and Wang (2005) present a new classifier, HARMONY, which is an example of direct mining for informative patterns as HARMONY directly mines the resultant set of rules required for classification. G. Dong and J. Li (1999) introduce a new type of patterns i.e. *emerging patterns* (EPs), for discovering knowledge from databases. They define EPs as data item-sets whose support increases more significantly from one to other data-set. They have used EPs to build very powerful classifiers. W. Fan et al (2008) builds a model based search tree, which partitions the data onto different nodes and at each node, it directly find out a discriminative pattern, which further divide its examples into more purer subsets. A novel technique was proposed by B. Liu, W. Hsu, and Y. Ma (1999), which first prunes the discovered association-rules to remove the insignificant association-rules from the entire set of association-rules, and then finds a subset of the un-pruned association-rules by which a summary of the discovered association-rules can be formed. They called that subset of association-rules as the *direction setting* (DS) *rules* because they can be used to set the directions, which are followed by the rest of the association-rules. By the help of the summary, the user can have more focus on the important aspects of the particular domain and also can view the relevant details. They suggest that their approach is effective as their experimental result shows that the set of DS rules is quite very small. Lent, Swami and J. Widom (1997) proposed a method for clustering two-dimensional associations in large data-bases. In their research work, they present a geometric-based algorithm called BitOp, for clustering, embedded within ARCS (Association Rule Clustering System). They also measure the quality of the segmentation generated by ARCS. J. Han et al (2006) proposed a new approach called CrossMine, which mainly includes a set of novel and powerful methods for multi-relational classification including 1) tuple ID propagation, 2) new definitions for predicates and decision-tree nodes and 3) a selective sampling method. They also proposed two accurate and scalable methods for multi-relational classification i.e. CrossMine-Rule and CrossMine-Tree. C. Zhang et al (2008) proposed a novel approach of combined patterns to extract important, actionable and impact oriented information from a large amount of association rules. They also proposed definitions of combined patterns and also design novel matrices to measure their interestingness and analyzed the redundancy in combined patterns. Combined mining as a general approach is proposed by C. Zhang et al (2011) to mine the informative patterns. They summarize general framework, paradigms and basic processes for various types of combined mining. They also generate novel types of combined patterns from their proposed frameworks. H. Yu, J. Yang and J. Han (2003) proposed a new method called as *Clustering-Based SVM* (CB-SVM), in which, they scan the whole data set only once to have an SVM with samples that carry the statistical information of the data by applying a hierarchical micro-clustering algorithm. They also show that CB-SVM is also highly scalable for very large data sets and also generating very high classification accuracy.

3. PROBLEM DEFINITION

As we know, complex data may contain incredible information, which cannot be mined directly just by using a single method, and also it is tough to deal with such information using different perspective such as client's perspective, business analyst's perspective and decision-makers perspective etc. as complexity arises. Any service provider wants to predict the client's behavior to design the services according to client's perspective and also to reduce the traffic load. In our approach, we try to get patterns to retrieve useful information from complex data. This information can be used in different places, for example in e-commerce, stock market, market campaigns, measuring the success of marketing efforts and client-company behavior etc.

4. PROPOSED SOLUTION

The effectiveness and quality of the patterns which have to be discovered highly depends on the richness and type of the data used during the pattern discovery process. We have taken our data-set from *UCI-Machine Learning Repository*, named as "Adult" for our project work. This data-set have 14 attributes, some of them are discrete attributes, while others are continuous attributes.

Our proposed solution according to the problem definition consists of following steps:

4.1 Preprocessing & Data mining approach

If any tuple in data-set contains unknown value (present as '?' in our data-set) for any of the attribute, then that tuple should be removed first, as such tuples are the source for noise and errors. So, we remove all such tuples from our data-set first. After preprocessing step, we make non-overlapping partitions of our dataset so that each of such partition can behave as a sub-dataset. The main idea behind generating sub-datasets is that then each of the sub-dataset can be used as a source of data for multi-source combined mining approach. We have generated 210 sub-datasets for our dataset as we try to have maximum number of distinct attributes from all of the sub-dataset on the basis of information gain value and we get maximum 8 distinct attributes (from $Figure\ 1$), when we generates 210 partitions of our dataset. Our analysis on the data-set for making the partitions is shown in the figure given below:

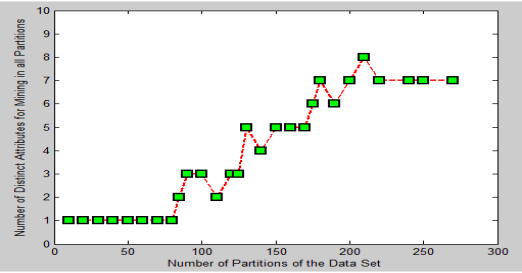

Fig. 1. Number of partitions vs. Number of distinct attributes

As we have already stated that our input dataset contains discrete as well as continuous attributes. We have computed the information gain value for discrete attributes in a partition P as given below:

We consider the expected information needed for the classification of a tuple in a partition P, if the class label attribute has n distinct values, each of which defines a distinct class[2], as follows:

$$\mathrm{I(P)} = -\sum_{i=1}^{n} \mathrm{p_i log_2 p_i} \quad (1)$$

Where, p_i is the probability that a particular tuple in partition P belongs to class C_i and we can compute p_i as $p_i = |C_{i,p}| \,/\, |P|$ and where $C_{i,p}$ defines the set of tuples of class C_i in P while $|C_{i,p}|$ and $|P|$ denotes the number of tuples in $C_{i,p}$ and P respectively.

Then we have to classify the tuples in P on some attribute A having m distinct values, as observed from the training dataset. Then the amount of information would be required for an exact classification [2] is measured by:

$$\mathrm{I_A}(P) = \sum_{j=1}^{m} (|\mathrm{P_j}|/|\mathrm{P}|) \times \mathrm{I(P_j)} \quad (2)$$

Then the information Gain $G(A)$ for attribute A is measured as:

$$G(A) = I(P) - I_A(P) \quad (3)$$

For computing the information gain for continuous attributes B in a partition P, we have to compute the information gain for every possible split-point for B and then choosing the best split-point. We consider split-point as a threshold on B. First, we have to sort the values of B in increasing order and then typically the mid-point between each pair of adjacent values considered as a possible split-point. So, if there are y values of B, then $(y-1)$ possible splits has to be computed. The reason for sorting the values of B is that if the values are already sorted then for determining the best split for B requires only one pass through the values. Then, for each possible split-point for B, by using $equation\ (2)$, we measured $I_A(P)$, where the number of value of $m = 2$. The point with the minimum expected information requirement for B will be selected as the best split-point for B.

After finding out the information gain for each attribute, we have taken an attribute with maximum information gain from each partition P_i and then by concatenating the attribute values from all partitions, we form a data-stream, which serves as an input for lossy-counting algorithm.

A. Lossy-counting Algorithm

Lossy-counting is a deterministic algorithm [2], which computes frequency counts over a stream of data-items. It approximates the frequency of items or item-sets within a user-specified error bound ε. If N is the current length of the data-stream then this algorithm takes $1/\varepsilon log(\varepsilon N)$ space in worst-case for computing the frequency counts of a single data-item. The steps for the implementation of this algorithm are as follows:

Input: Support s, error bound ε and input data-stream

Output: Set of data-items with frequency counts at least equals to $(s - \varepsilon)N$

Step 1: The input data-stream logically divided into the buckets of width $w = ceil(1/\varepsilon)$ and each bucket is labeled with bucket id, initially starting from 1 for the first bucket, and the current bucket id is denoted by $B_{current}$, which is equal to $ceil(N/\varepsilon)$.

Step 2: Then maintain a data-structure DS, which is a set of values of the form (E, F_E, δ), where, E is an element from the input data-stream and F_E is the true frequency of the element E and δ

denotes the maximum number of times E could have occurred in first $B_{current} - 1$ buckets. Initially DS will be empty.

Step 3: For an element from the data-stream, if E already exists in DS then increase its F_E by 1 else we have to create a new entry in DS such as $(E, 1, B_{current} - 1)$.

Step 4: If it is the bucket boundary then we have to prune DS as follows:
if $F_E + \delta \leq B_{current}$, then the entry (E, F_E, δ) has to be deleted from DS.

Step 5: When a user wants a final list of frequent data-items with support s, then output all those entries in DS with $F_E \geq (s - \varepsilon)N$.

Then, we generate the combined association rules (C. Zhang et al, 2011) of the frequent data-items computed by lossy-counting algorithm.

4.2 Multi-feature mining approach

After the generation of the combined association rules, we consider the heterogeneous features of different data types as well as of different data categories.

If the combined association rule is of the form "$IF\ X\ THEN\ Y$", where X is the antecedent and Y is the consequent part of the rule, then we have some traditional definitions for support, confidence and lift of the rule as given below in the Table 1.

$SUPPORT$	$Prob(X\ U\ Y)$
$CONFIDENCE$	$Prob(X\ U\ Y)\ /\ Prob(X)$
$LIFT$	$Prob(X\ U\ Y)\ /\ (Prob(X)\ \times\ Prob(Y))$

Table 1. Support, Confidence and Lift for the Rule $X \rightarrow Y$

On the basis of these traditional definitions of support, confidence and lift, we can compute the Contribution and Interestingness [1] $I_{RULE}$of the rule $X_p U X_R \rightarrow Y$ as follows:

$$Contribution\ (X_P U X_R \rightarrow Y) = LIFT\ (X_P\ U\ X_R \rightarrow Y)\ /\ LIFT\ (X_P \rightarrow Y)$$
$$= CONFIDENCE\ (X_P\ U\ X_R \rightarrow Y) / CONFIDENCE\ (X_P \rightarrow Y) \quad (4)$$
$$I_{RULE}\ (X_P U\ X_R \rightarrow Y) = Contribution\ (X_P\ U\ X_R \rightarrow Y)\ /\ LIFT\ (X_R \rightarrow Y) \quad (5)$$

Where, I_{RULE} indicates whether the Contribution of X_P (or X_R) to the occurrence of Y increases, while considering X_R (or X_P) as a precondition to the rule. To get more information, we also generate pair patterns, cluster pattern, incremental pair-pattern and incremental cluster-patterns (C. Zhang et al, 2011), and their respective contribution and interestingness matrices. In case of pair pattern, two atomic rules are taken to form a pair-pattern if and only if the two atomic rules have at least one common data-item in their antecedent parts and after removing those common data-item/data-items from the atomic rules the antecedent parts of none of the atomic rules should be null. In case of incremental pair-pattern, we actually remove the common data-item/data-items from the pair-patterns and consider the common data-items as a pre-condition. In case of cluster pattern formation, we try to include as much as rules in a single cluster on the basis of the common data-item/data-items in their antecedent parts and also take care of the fact that after removing the common data-item/data-items from the antecedent parts of the respective rules, the antecedent part of the rules should not be empty or null. In case of incremental cluster-patterns, we actually remove the common data-item/data-items from the respective rules in a cluster and consider the common data-item/data-items as a precondition for that particular cluster of rules.

4.3 Multi-method mining approach

In this approach, we first generates the association rules by using FP-growth algorithm(Han and Kamber, 2006) and then make the bayesian belief network (Jie Cheng, D. A. Bell and Weiru Liu, 1997) by those association rules during training phase and then classify the testing data-set by Bayesian belief network during testing phase.

5. RESULT AND ANALYSIS

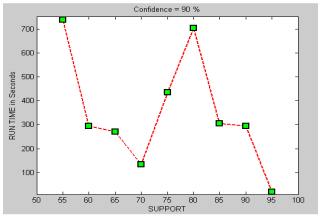

Fig. 2. Support vs. Run-Time

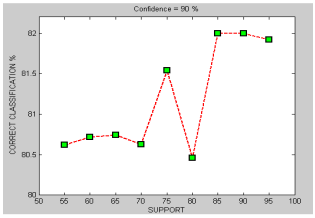

Fig. 3. Support vs. Correct Classification %

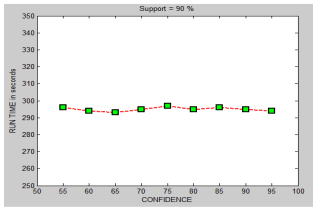

Fig. 4. Confidence vs. Run-Time

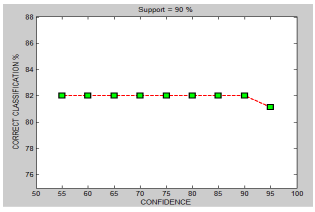

Fig. 5. Confidence vs. Correct Classification %

We take our data-set (as mentioned earlier) with 32,561 entries and after preprocessing step, we get 30,162 entries, for mining the useful information. We have used Java as a programming language interface. By getting various kinds of patterns, we mine the information by including other perspective like client's perspective etc. In $fig.2$ & 3, we have shown the variation of run-time (in seconds) and correct classification percentage by bayesian belief network vs. support by keeping $confidence = 90\%$ fixed for FP-growth implementation and in $fig.4$ & 5, we have shown the variation of run-time (in seconds) and correct classification percentage by bayesian belief network vs. confidence by keeping $support = 90\ \%$ fixed for FP-growth implementation.

6. CONCLUSION AND FUTURE WORK

The support have impact on run-time as well as correct classification percentage by Bayesian belief network, while confidence have approximately no effect neither on run-time nor on correct classification percentage. We have also identified combined patterns, which are more informative, actionable and impact-oriented as compared to any single patterns identified by traditional methods like FP-growth etc. We can have such frameworks, which are flexible and customizable for handling a large amount of complex data, for which data sampling and table joining may not be acceptable. We further can develop some effective paradigms, for handling large and multiple sources of data available in industry projects for government, insurance, stock market, e-commerce and banking etc. in real-time.

REFERENCES

[1] C. Zhang, D. Luo, H. Zhang, L. Cao and Y. Zhao (2011), 'Combined Mining: Discovering Informative Knowledge in Complex Data', IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 41, NO. 3, pp. 699-712.

[2] Han and Kamber (2006), 'Data Mining Concepts and Techniques', 2nd ed., United State of America.

[3] C. Zhang, F. Figueiredo, H. Zhang, L. Cao and Y. Zhao (2007), 'Mining for combined association rules on multiple datasets', in Proc. DDDM, pp. 18–23.

[4] C. Zhang, H. Bohlscheid, H. Zhang, L. Cao and Y. Zhao (2008), 'Combined pattern mining: From learned rules to actionable knowledge', in Proc. AI, pp. 393–403.

[5] B. Park and H. Kargupta (2002), 'Distributed Data Mining: Algorithms, Systems, and Applications'. Data Mining Handbook, N. Ye, Ed 2002.

[6] Jaturon Chattratichat, John Darlington, et al (1999). 'An Architecture for Distributed Enterprise Data Mining'. Proceedings of the 7th International Conference on High-Performance Computing and Networking.

[7] B. Park, D. Hershbereger, E. Johnson and H. Kargupta (1999), 'Collective data mining: A new perspective toward distributed data mining'. Accepted in the Advances in Distributed Data Mining, Eds: Hillol Kargupta and Philip Chan, AAAI/MIT Press (1999).

[8] G. Dong and J. Li (1999), 'Efficient mining of emerging patterns: Discovering trends and differences,' in Proc. KDD, pp. 43–52.

[9] H. Cheng, J. Han, J. Gao, K. Zhang, O. Verscheure, P. Yu, W. Fan and X. Yan (2008), 'Direct mining of discriminative and essential graphical and item-set features via model-based search tree,' in Proc. KDD, pp. 230–238.

[10] B. Liu, W. Hsu and Y. Ma (1999), 'Pruning and summarizing the discovered associations,' in Proc. KDD, pp. 125–134.

[11] A. N. Swami, B. Lent and J. Widom (1997), 'Clustering association rules,' in Proc. ICDE, pp. 220–231.

[12] J. Han, J. Yang, P. S. Yu and X. Yin (2006), 'Efficient classification across multiple database relations: A CrossMine approach,' IEEE Trans. Knowl. Data Eng., vol. 18, no. 6, pp. 770–783.

[13] C. Zhang, H. Bohlscheid, H. Zhang, L. Cao and Y. Zhao (2008), 'Combined pattern mining: From learned rules to actionable knowledge,' in Proc. AI, pp. 393–403.

[14] H. Yu, J. Han and J. Yang (2003), 'Classifying large data sets using SVM with hierarchical clusters,' in Proc. KDD, pp. 306–315.

[15] Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

[16] David A. Bell, Jie Cheng and Weiru Liu (1997), 'An Algorithm for Bayesian Belief Network Construction from Data,' In Proceedings of AI & STAT'97, pp. 83-90.