Approximate Functional Dependencies Mining Using Association Rules Specificity Interestingness Measure

Jalal Omer Atoum[1*]
[1]Computer Science Department, Princess Sumaya University for Technology, Amman, Jordan

ABSTRACT

Mining Approximate Functional Dependencies (AFDs) from a database may produce valuable interesting relationships among its variables that would be beneficial in several domain applications such as marketing, financial data analysis, biological data analysis, and intrusion detection. Mining of association rules is concerned with extracting new knowledge from databases in the form of patterns and associations among data items. The mining of AFDs still posing special set of challenges such as the space and time requirements in addition to the quality of the discovered AFDs. In this paper, an approach for AFDs mining is being developed through employing the specificity interestingness measure and its monotonic property used in some association rules mining algorithms. This approach has been tested on a set of test bed of datasets. The results showed an improvement in time requirements and in the number of mined AFDs.

Keywords: Approximate Functional Dependencies; Mining AFD; Interestingness Measures; Specificity measure.

1 INTRODUCTION

Functional dependencies (FDs) are useful knowledge that are being mined from databases. Such knowledge has been represented in the form of patterns that hold between some database variables on all of its tuples [11]. When there are some tuples that do not satisfy such a pattern, the mined FD is called approximate FD (AFD). Mining of AFDs from databases in which there are some expected relationships among the database variables but with the existence of errors due to various reasons that prohibits the discovery of such FDs has become very important. This is true in several applications and domains such as marketing, market basket analysis, web usage mining, intrusion detection, and bioinformatics [14]. Association rules are conditional statements that point out or discover relationships among data items in a database.

Mining association rules is the detection of patterns among data items in a database [15]. The purpose here is to identify rules among various data items in a database that have strong measures of interestingness. For example, the association rule:

Milk, Fruit → Soda

Found from a supermarket database transactions would indicate that when a customer buys milk and fruit, he would most likely buy soda.

Hence, association rules are dependencies that exists between particular values of some attributes. Whereas, approximate functional dependencies AFDs) are dependencies that are true to some extent on the entire database (i.e. it uses attribute names instead of attribute values). For example, the following AFD:

Language → Nationality

May hold on most of the tuples of a database.

Furthermore, an AFD can be considered as a condensed form of several association rules [13]. For example, given the following set of association rules:

English → British
Urdu → Pakistan
Hindi → India
Greek → Greece.

Would be condensed into the following AFD:

Language → Nationality

Which is represented by the graph shown in Figure 1.

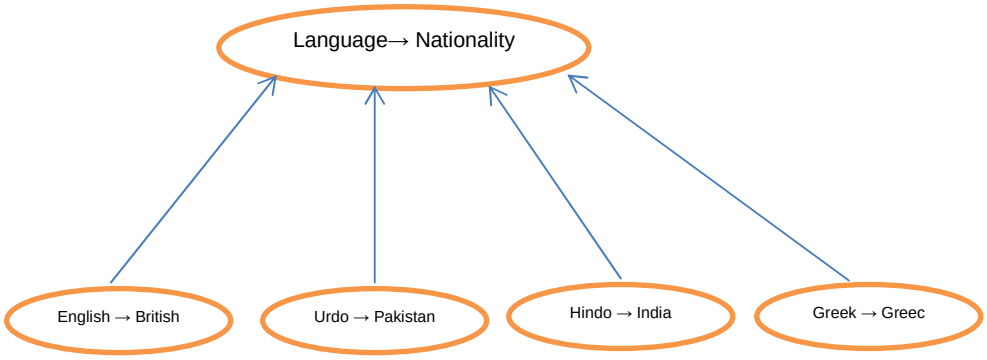

Fig. 1. Condensed Association Rules Example

Section two discusses AFDs mining issues specifically those related to performance and interestingness. Section three presents the AFDMCEC Algorithm in which our work is based on. Section four presents our proposed approach of mining AFDs. Section five presents the evaluation of our approach. Finally, section six provides the conclusion.

2 AFDs MINING ISSUES

The most two important issues that face AFDs mining algorithms are the performance issue in terms of space and time and the interestingness measures issue that determines the quality of mined AFDs.

2.1 PERFORMANCE ISSUE

Most mining techniques of AFDs from databases suffer from the high cost associated with them. Such costs are due to the fact that pruning strategies used in mining functional dependencies are not suitable for mining AFDs. Pruning strategies in FD mining techniques usually find minimal covers set of FD and may eliminate some interesting AFD [12]. Furthermore, the search space needed to discover AFDs from a large database with large number of attributes is very huge. Finally, the determination of whether an FD hold or not is very costly since we may need to test such an FD against each transaction in the database [1 and 9].

The aim of the pruning process is to reduce the search space of uninteresting parts, taking into account the guidelines of not losing any interesting AFD. If there should arise an occurrence of AFDs, negligible conditions help us pruning a lot of the space needed. One of the most difficult challenges is finding a good interestingness measure that would be effective in pruning the search space during the mining process of AFDs. Also, this cost can be reduced by selecting the appropriate approximation measure and/or selecting some constrains that would prune AFDs and their subsets from being considered.

The traversing of space in rule mining procedures are ordinarily done either through a top down or a bottom up approach using all combination in the attributes lattice of a database [7]. In general, the traversal process is affected by both the pruning rules and the interestingness measures used in discovering ADFS. Hence, it is not straight forward to select either bottom up or top down as a traversing approach.

2.2 INTERESTINGNESS MEASURE

One of the major problems of data mining is that most algorithms produce a large set of AFD or association rules that would be of no interest to the end user and be very hard to capture or to analyze. Here, we will be facing the problem of quantity against the quality, especially for the class of Apriori data mining algorithms that use the support-confidence measures [3].

An approach that would help alleviate such a dilemma is to involve the end user in this process. Visual data mining that adopt the human centered approaches and emphasize the interaction between the user and the data mining algorithms would be worth more investigation and evaluation. Moreover, data mining algorithms usually use interestingness measures in order to filter out discovered knowledge (association rules or AFDs).

Several techniques have been used to measure the interestingness or the suitability of associations rules[6 and 16] . Such measures should be evaluated in terms of the applicability of the mined association rules to be presented to the user. In other words, would such mined association rules be applicable and anticipated to the user. Hence, such measures would be subjective as only the user of such rules can determine their interestingness.

For an AFD mining technique to be practical it should use some interestingness measures. Therefore, some guidelines must be characterized to evaluate the interestingness measures based on the application in which they are proposed to be utilized. Like the determination of an appealing of standard within KDD, an issue is the manner by which to gauge the extent to which a practical reliance (FD) is estimated. Furthermore, the monotonic property of approximation measures is always being applied by any extraction of AFDs regardless of the approximation measure used. This means that if an AFD satisfies an interestingness measure, we do not need to check any other of its superset of AFDs [3].

Moreover, objective measures of association rules such as; Support, Confidence or strength, Conviction, Lift, Interest, Correlation, Leverage, Piatetsky-Shapiro and Coverage are also of interest [6]. However, such measures may provide conflicting information regarding their interestingness. In addition, the performance of some interestingness measures may depend on the application domain.

For an association rule of the form A → B, its confidence means that the conditional probability of B given A. Whereas, for the AFD X → Y, the confidence means the chance of finding Y-value given the X-value. This requires, finding the association rule that has the highest confidence value for every distinct pair values of X and Y.

Also, for an association rule of the form A → B, its support value means the probability that event A occurs. An association rule with high confidence value and low support value is consider to be an uninteresting rule [8].
In general, when there are several association rules with low support values, the discovered approximate FD from such rules will be useless and of no interest to users. For example, this will have low predication rates in classification applications and few query output in query rewriting applications [17].

The support values of a set of condensed association rules would have a great impact on the quality of their corresponding AFD. Regardless of high values of confidence, low values of support make such association rules less interesting to users and consequently produce bad AFDs. For example, the following AFD:

Nationality → Language

Would have a high confidence with low support, therefore such an AFD will be useless and of no interest to users. However, for the following AFD:

Major, Level → Fees

Would have few association rules with high support and many with low support, such an AFD would be of an interest to the user.

The specificity of an association rule is considered to be an interestingness measure that assesses the percentage of negatives that are correctly identified [4]. For example, in a medical test for diagnosing a disease, the percentage of correctly detected patients without a condition whom have negative test results for the particular disease can be defined as follows:

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}$$

$$= \frac{\text{number of true negatives}}{\text{total number of well individuals in population}}$$

$$= \text{probability of a negative test given that the patient is well}$$

The Specificity measure is based on the information entropy and it should allow users to determine what type of AFDs that would be of interest to them. This imply that a higher value of specificity would result in an uninteresting discovered AFDs.

Hence, for an AFD X → A to be interesting to a user, the database would have few distinct values of X and several tuples for the same value of X. The distribution of X values can be computed using the specificity measure. For a database of N tuples, the specificity of attribute X is defined in terms of the support of every value of X { $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ divided by the maximum entropy that is defined when X is a key and it is equal to $\log_2(N)$ [5].

$$\text{Specificity (X)} = -\frac{\sum_1^m support(\alpha 1)\ x\ log2(support(\alpha 1))}{log2(N)}$$

The possible values of Specificity (X) are between 0 and 1, inclusive. When there is only one possible value of X , this would result in a maximum support , and would consequently results in a least value of specificity, namely a value of zero. However, when all of the values of X are distinct, X is a key for the database, least support would be resulted, and consequently results in a maximum value of specificity, namely a value of one. Therefore, the lower the specificity of X in the AFD (X → A) , the more interesting this AFD would be to users. Whereas, as the specificity(X) increases, the less interesting the AFD would be to users.

The will lead us to the Monotonicity property of Specificity, that indicates for any two attribute sets X and Y, such that X ⊃ Y, always specificity(X) ≤ Specificity(Y). This would be used as a pruning strategy to reduce the search space of association rules in the mining process of AFDs from a database.

Finally, for a user specified values for minimum support (minsupp) and maximum specificity (maxspec), an ADF mining algorithm would be searching for all AFDs of the form (X → A) such that confidence(X → A) ≥ minsupp and specificity(X) ≤ maxspec. This Monotonicity property of Specificity and the combination of confidence and specificity measures are used in our proposed mining approach presented in section 4.

When compared to other measures such as sensitivity, error rate and accuracy, the specificity measure produces better quality AFDs and more effective in reducing the search space during the level-wise search for candidate generations.

3 AFDMCEC ALGORITHM

The work presented in this paper is based on a previous ADF extraction algorithm called Approximate Discovery of Functional Dependency using Minimal Cover and Equivalent Classes (AFDMCEC)[2]. This algorithm discovers all AFDs from databases based on the approximation measure g_3 and on the employment of the concepts of equivalent properties and minimal (Canonical) cover of FDs. The g_3 measure is defined to be the minimum number of tuples that have to be removed from a database for an FD to hold divided by the number of tuples in this database.

The results obtained from this algorithm had shown a better time requirements when compared with a modified version of TANE algorithm [7] using the g_3 approximation measure. TANE algorithm finds all non-trivial FDs by searching the lattice in a level-wise manner. Whereas, g_3 is an error rate that is defined to be the minimum number of rows to be removed from a relation instance for an FD to hold. For more detail of the AFDMCEC algorithm refer to [2].

4 PROPOSED ALGORITHM

The proposed modification of the AFDMCEC Algorithm employs the Monotonicity property of Specificity and the combination of confidence and specificity interestingness measure in mining rule associations techniques that would be condensed into AFDs. This modification aims to improve the quality of extracted ADFs and to reduce the cost associated with extractions.

In particular, the proposed modified approach involves a pruning strategy that is based on a computed confidence value and consequently the specificity value of each association rule that would be condensed into AFDs. The actual pruning is done through trimming the search space by eliminating all association rules if their specificity values are less than a predetermined threshold and consequently elimination all of the association rules whose left hand side are supersets of the eliminated association rule (the Monotonicity property of Specificity measure). Hence, for any two attribute sets X and Y, such that $X \supset Y$ then specificity(X) ≤ Specificity(Y). Such property would reduce the search space of association rules during the mining process of AFDs from a database

In addition, as it had been used in [2], all keys and all of their supersets are removed during the pruning process. Also, for each pair of attributes that have equivalent closures, one of these attributes is removed from the candidate sets of attributes during the search process. The pruning strategy used is presented in the following procedure:

```
Procedure PRUNE(L`)
   for each X ∈ Lℓ do
         if C+ (X) = Θ do delete X from L ℓ
         if X is a (super)key do
                  for each A ∈ C+ (X) \ X do
                           if A ∈ ∩ b∈X C+ (X ∪ {A} \ {B}) then output X → A
                  delete X from Lℓ
         if  Specificity (X) ≥ maxSpecificity then  delete X from L`
```

5 EXPERMINTAL ANALYSIS

To evaluate the performance of the proposed algorithm (Denoted by PropADF), an experimental tests had been conducted to extract ADFs from a test bed of datasets called UCI [10]. Also, the same experiments had been conducted on the previous AFDMCEC Algorithm (Denoted by PrvADF). This dataset, as shown in Table1, contains databases with varying number of attributes and varying number of tuples.

Table 1: Test Bed Dataset.

Database Name	# of attributes	#of Tuples
Abalone	9	4177
Balanace-scale	5	625
Breast-cancer	11	699
Bridge	13	108
Chess	7	28056
Echocardiogram	13	132
Glass	11	214
Iris	5	150
Nursery	9	12960

Table 2 presents the time requirements in seconds to discover all AFDs from a database using the previous AFDMCEC mining Algorithm and our proposed mining approach with varying threshold values of ε (g_3) and maxspefic varying from 0.05 to 0.75.

Table 2. CPU Times Requirements To Mine AFDs from Data set.

DataBase	ε = maxspefic =0.05		ε = maxspefic = 0.25		ε = maxspefic = 0.50		ε = maxspefic = 0.75	
	PrvAFD	PropAFD	PrvAFD	PropAFD	PrvAFD	PropAFD	PrvAFD	PropAFD
Abalone	0.84	0.72	0.27	0.25	0.09	0.085	0.073	0.089
Balanace-scale	0.01	0.013	0.008	0.009	0.004	0.042	0.003	0.007
Breast-cancer	13.43	11.62	3.12	3.02	1.08	1.072	1.04	1.21
Bridge	83.52	78.12	10.04	8.24	3.17	3.02	3.04	3.50
Chess	0.08	.07	0.016	0.014	0.012	0.01	0.011	0.042
Echocardiogram	5.04	3.29	1.24	1.07	0.18	0.16	0.168	0.19
Glass	0.53	.48	0.09	0.08	0.07	0.06	0.063	0.075
Iris	0.004	0.01	0.001	0.007	0.001	0.002	0.001	0.003
Nursery	2.36	0.36	0.73	0.59	0.16	0.142	0.125	0.173

Figure 2 presents graphically the average time in seconds needed to discover all AFDs over all datasets for each algorithm over the same range of maxspefic.

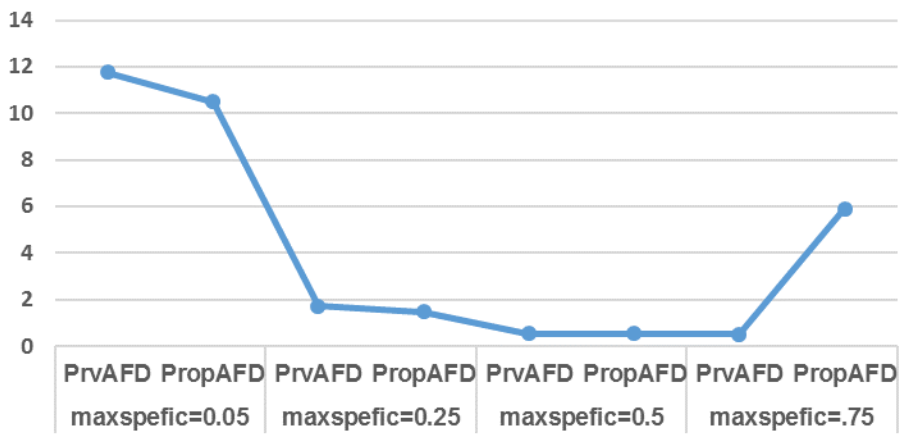

Fig. 2. Average Time in Seconds Needed To Discover all AFDs from All Datasets

It is worth mentioning that different environment and machine are used in this current research from the those used previously in the AFDMCEC Algorithm.

Table 3 presents the number of AFDs that had been checked by each algorithm with varying threshold values of ε (g_3) and maxspefic from 0.05 to 0.75.

Table 3. Number of checked AFDs from Data set.

DataBase	ε = maxspefic = 0.05		ε = maxspefic = 0.25		ε = maxspefic = 0.50		ε = maxspefic = 0.75	
	PrvAFD	PropAFD	PrvAFD	PropAFD	PrvAFD	PropAFD	PrvAFD	PropAFD
Abalone	517	483	457	418	407	319	542	563
Balanace-scale	62	65	56	59	42	48	63	67
Breast-cancer	4725	4517	4346	4163	4037	3926	4837	4942
Bridge	13254	12418	11975	11053	10274	9825	14620	14875
Chess	387	347	315	293	286	275	408	402
Echocardiogram	2379	2175	2042	1874	1752	1581	2475	2512
Glass	335	297	317	252	297	236	386	394
Iris	64	67	52	61	43	56	65	68

Nursery	1872	1543	1624	1312	1427	1038	2171	2275

Figure 3 presents graphically the average number of checked AFDs over all datasets for each algorithm over the same range of maxspefic.

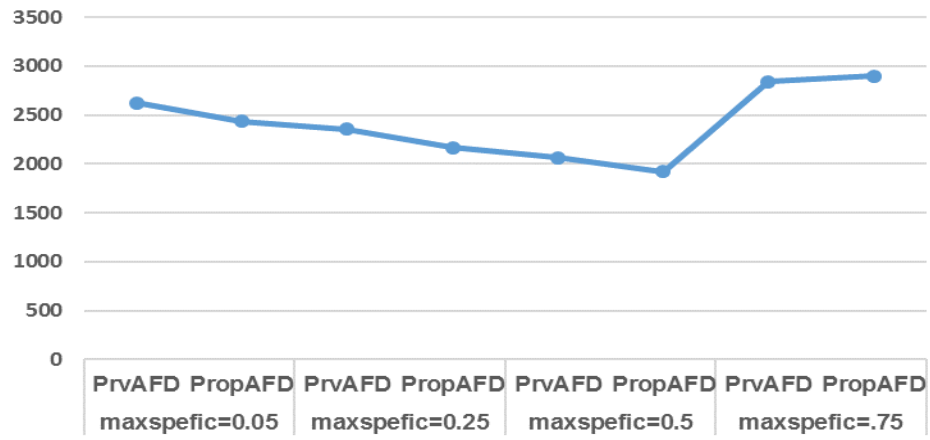

Fig. 3. Average Number of AFDs Checked for all Datasets

It is clear from Table 3 and Figure 3 that the minimum (best) number of checked AFDs is reached when the maxspefic is equal to 0.5. For the threshold value of 0.75, the number of checked AFDs and the time requirements have increased since more uninteresting AFDs have been checked as a consequence of less pruning is being performed during the mining process.

Furthermore, it can be noticed from Table 3 and Figure 3 that for low threshold values of ε (g_3) and maxspefic (0.5, 0.25, and 0.5), the number of AFDs that had been checked by the proposed mining approach is less than those checked by the previous algorithm. However, for a large threshold value (0.75) the number of AFDs that for low threshold values of ε (g_3) and maxspefic (0.5, 0.25, and 0.5), the performance of the proposed approach (PropAFD) is much better than the previous algorithm (PrvAFD). It is also clear that the best performance to mine all AFDs is reached when the maxspefic is equal to 0.5. However, for a large threshold value (0.75), the performance of the proposed approach is worse than that of the previous algorithm. This is due to the fact that as the value of the specificity increases it would discover more an uninteresting AFDs and would have less pruning. However, for databases with low number of attributes, the performance of the proposed approach is worse than the previous algorithm. Hence, our proposed mining approach would perform better with databases that have large number of attributes.

Notice that the current results of the previous algorithm is slightly different from what had been reported in [2]. This is due to the
that had been checked is more than those checked by the previous algorithm. Again, this is due to the fact that as the value of the specificity increases it would discover more an uninteresting AFDs and would have less pruning.

6 CONCLUSION

In this paper we have studied how AFDs and association rules are related to each other. Some challenges issues of mining association rules and consequently AFDs had been discussed. The specificity interestingness measure and its monotonic property used in

association rules mining algorithms have been illustrated especially their effects in mining quality association rules. A modified version of a previous mining of AFD had been developed and implemented in which it employs this specificity interestingness measure as the main driving pruning strategy to reduce the time mining requirements and to produce a better quality AFDs.

The proposed AFDs mining approach have been tested on test bed dataset, for various threshold values of specificity. The result obtained showed that this proposed AFDs mining approach is better in terms of time and in number of AFDs produced from those obtain from the previous algorithm especially for low threshold values of specificity.

Competing Interests

Author has declared that no competing interests exist.

Acknowledgment

I would like to thank Princess Sumaya University for Technology for supporting this research through my sabbatical year.

REFERENCES

[1] Abedjan, Z., Schulze, P., and Naumann, F., "DFD: Efficient Functional Dependency Discovery", In Proceedings of the International Conference on Information and Knowledge Management (CIKM), pages 949–958, 2014.

[2] Atoum, J., "Mining Approximate FDs from Databases Based on Minimal Cover and Equivalent Classes," European Journal of Scientific Research, 33(2), pp. 338-346, 2009.

[3] Agrawal, Rakesh; and Srikant, Ramakrishnan, "Fast Algorithms for Mining Association Rules in Large Databases", *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September*, pp. 487-499. 1994.

[4] Geng, L., and Hamilton, H., "Interestingness Measures for Data Mining: A Survey", ACM Comput Survey, 38(3):1– 32, 2006.

[5] Gummadi, R., Khulbe, A., Kalavagattu, A., Salvi, S., and Kambhampati, S., "SmartInt: Using Mined Attribute Dependencies to Integrate Fragmented Web Databases", In: Proceedings of the 2011 International Conference on the World Wide Web , 2011, . pp. 51-52, 2011.

[6] Hahsler, M.,"A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules", http://michael.hahsler.net/research/association_rules/measures.htm, 2015.

[7] Huhtala, Y., Kärkkäinen, J., Porkka, P., and Toivonen, H.,TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies. The Computer Journal, 42(2), pp 100-111, 1999.

[8] Li, J., "Research on Association Rule Mining Algorithm D", *Xi'an* University of Electronic Science and Technology, 2008.

[9] Li, J., Liu, J., Toivonen, H., and Yong, J, "Effective Pruning for the Discovery of Conditional Functional Dependencies", *The Computer Journal* 56 (3): 378-392. 2013.

[10] Lichman, M., "UCI Machine Learning Repository", Irvine, CA: University of California, School of Information and Computer Science. 2013, http://archive.ics.uci.edu/ml.

[11] Liu, J., Li, J., Liu, C., and Chen, Y., "Discover Dependencies from Data – A Review", IEEE Transactions on Knowledge and Data Engineering (TKDE), 24(2):251–264, 2012

[12] Lopes, S., Petit, J., and Lakhal, L., "Functional and Approximate Dependency Mining: Database and FCA Points of View", J. Exp. Theor. Artif. Intell.,14(2-3):93–114, 2002.

[13] Mannila, H. and Toivonen, H., "Multiple Uses of Frequent Sets and Condensed Representations", In Proceedings of the KDD Conference, pp. 189–194, 1996.

[14] Nakod, N., and Vaidya, M., "Survey on Multidimensional and Conditional Hybrid Dimensional Association Rule Mining", International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, 1(4), pp 63-66, 2013.

[15] Piatetsky-Shapiro, Gregory, "Discovery, Analysis and Presentation of Strong Rules", *in* Knowledge Discovery in Databases*, pp*. 229-248, 1991.

[16] Tan, P., Kumar, V., and Srivastava, J., "Selecting the Right Objective Measure for Association Analysis", Information Systems, 29(4):293-313, 2004.

[17] Wolf, G., Khatri, H., Chokshi, B., Fan, J., Chen, Y., and Kambhampati, S., "Query Processing over Incomplete Autonomous Databases". In VLDB '07: Proceeding of the 33rd international conference on Very Large Data Bases , pp 651-662. VLDB Endowment, 2007.