Enhanced Associative classification based on incremental mining Algorithm (E-ACIM)

Mustafa A. Al-Fayoumi

College of Computer Engineering and Sciences, Salman bin Abdulaziz University
Al-Kharj, Saudi Arabia

Abstract

Several association classification algorithms have been designed to build efficient and accurate associative classifiers for large data set . Most of classification algorithms are still suffer from the huge number of the generated classification rules which takes efforts to select the best ones in order to construct the classifier. Moreover, When different data operations (adding, deleting, updating) are applied against certain training data set, the majority of current AC algorithms must scan the complete training dataset again to update the results (classifier) in order to reflect changes caused by such operations. To overcome such drawbacks, the Associative Classification based on Incremental Mining (ACIM) algorithm has been proposed to handle with incremental data without jeopardizing the classification accuracy. The main assumption of ACIM algorithm is to build a classifier when new data arrive by utilizing the old classifier without rescanning to the entire dataset. This paper introduced a modified ACIM algorithm called Enhanced ACIM (E-ACIM). This algorithm deals with the data insertion problem in associative classification context. The E-ACIM is competitive and more efficient in computational time compared with ACIM and CBA algorithms and almost provides the same accuracy for both algorithms.

Keywords: *Associative classification, CBA, ACIM, Incremental learning.*

1. Introduction

Recent developments of information technology and computer networks caused the production of large quantities of databases. These databases normally contain hidden useful information that can be utilized in decision making and corporate plans. Therefore, efficiently finding and managing the useful information from these large databases become a necessity. One of the common tools which discovers and extracts non obvious knowledge from different types of data is data mining. The importance of data mining is growing rapidly in recent years since it can be used for several different tasks including classification, clustering, regression, association rule discovery, and outlier analysis [1].

The mining of association rules on data is usually an offline process since it is costly to find the association rules in large databases. Moreover, in most real world applications like stock market exchange, online transaction, retail marketing, and banking, data usually are updated on a daily basis, as well as, new data are generated and old data may be obsolete with the progress of time. As a result, efficient incremental updating algorithms are required for maintenance of the discovered association rules to avoid redoing mining on the whole updated database and therefore handling the incremental learning problem becomes crucial in these applications [2, 3].

In spite of many mining algorithms for discovering frequent data item and associations have been introduced, the process of updating frequent item sets is still disturbing to incremental learning algorithms. The process of updating the classifier in order to reflect the effect of incremental data is complicated and may lead to rescanning the original dataset for checking whether the itemsets stay frequent while new data are added.

This paper presents a new technique to reduce the computation time for rebuilding the classifier in the ACIM algorithm by using the stratified random sample of incremental data as a testing data to evaluate the classifier derived from the original training dataset. In the case of the accuracy for the classifier on the testing data sample is less than the derived classifier from the original data then a new classifier are constructed again. The new technique called Enhanced Associative Classification based on Incremental Mining (E-ACIM).

The remaining of the paper is organized as follows: the related work is presented in section 2. Section 3, describes the framework of the proposed Enhanced Associative Classification based on Incremental Mining (E-ACIM). The algorithm details are presented in section 4. The experimental results are demonstrated in section 5. The paper is conclude in section 6.

2. Related Work

Incremental learning is one of the challenges related to data mining tasks, especially association rule and

classification. In classification context, the problem involves updating the classification model (classifier) when the training data collection gets updated.

In association rule discovery several incremental algorithms have been developed such as Fast Update (FUP) [4], FUP2 [5], Insertion, Deletion and Updating [6], Galois Lattice theory [7], and New Fast Update (NFUP) [8]. However, in classification especially associative classification [9] and rule induction [10] scholars have paid little attention to the incremental learning issue. Furthermore, since classification is a common task in data mining and has various number of important applications in which data are often collected by these applications on a daily, weekly or monthly. Consequently, the efficiency and the effectiveness of techniques for incremental mining are both crucial issues and there is a great important to develop or enhance the current classification methods to deal with the incremental learning problem.

In the last few years an approach that integrates association rule and classification called Associative Classification (AC) was proposed [9]. Several research studies [9, 11, 12] provide evidence that AC often successfully builds more accurate classifiers than traditional classification approaches such as rule induction and decision trees. Furthermore, many application domains including image analysis [13] and document classification [14] have adopted AC since it generates simple "IF-THEN" rules that are easy to interpret by end-user.

Associative classification (AC) is an approach in data mining that integrates association rule discovery into classification problems to build classification-n systems that are easy to interpret by end-user. One of the first algorithms to deal with that integration is CBA algorithm which was proposed by Liu et al. in 1998 [9]. The CBA consist of two phases: a rule generator (CBA-RG) algorithm for discovering candidate class association rules (CARs), and a classifier builder (CBA-CB) for choosing a set of high precedence rules from the completed set of CARs to cover the training data.

The rule generator (CBA-RG) follows the basics principles of the Apriori algorithm in order to generate frequent ruleitems. Therefore the problem is reduced to the discovery of the frequent 'ruleitems' because this approach requires repetitive scanning over the database, which increases the required processing time by finding frequent ruleitems from all possible candidates at each level. Finding frequent ruleitems from all possible candidate ruleitems at each level is considered as a bottleneck of Apriori algorithm.

In AC algorithms, when new data added to original dataset, these algorithms could deal with this issue by three ways: Firstly, using the current classifier without any consideration to incremental dataset (d+) to classify unseen data, thus the classifier will not reflect the changes on the dataset. Consequently, the accuracy of classifier will be decreased, because a lot of previous CARs (class association rule) of classifier become invalid after dataset updating. Secondly, rebuilding the classifier from scratch depending on the original dataset (***D***) and the incremental dataset ($\boldsymbol{d}^+$) (i.e. D U d^+), but this approach require a complete scan to the whole training data set in order to reflect the new changes on the classifier. This means regenerating most of the CARs that were produced in the previous scan. Now, since new rules are generated and existed rules may be discarded after a data operation is executed against the training data set, which definitely causes time overhead. Thirdly, using the AC based on incremental mining algorithms like **ACIM** algorithm [15]. The ACIM deal only with a row insertion aspect of the incremental learning.

The E-ACIM algorithm is designed based on ACIM which is type of association classification algorithms, in this section a brief summary of ACIM is introduced as the base of E-ACIM. The ACIM mainly consists of the following three steps to build final classifier:

Step 1: Association rule mining for generating a set of class association rules (CARs).

Step 2: The pruning process for discarding the redundant and harmful rules. The high quality rules hold to construct the final classifier.

Step 3: Prediction process for classifying the new data object (unseen data) by classifier.

ACIM processes depend on two important definitions : Support and Confidence. ACIM extracts all strong rules (CARs) that are satisfied minimum support and confidence threshold of the increment data ($\boldsymbol{d_i}$) using Apriori Algorithm like CBA. The minimum support and confidence threshold of $\boldsymbol{d_i}$ is similar to original minimum support and confidence. The matching between the new CARs that are generated from the $\boldsymbol{d_i}$ and the current classifier is carried out for updating the support and confidence of the CARs that exist in the current classifier and frequent in the $\boldsymbol{d_i}$. Moreover, identical CAR or general rule exist in the classifier will be removed.

The CARs that are generated from $\boldsymbol{d_i}$ but not exist in the classifier will be checked against original database for removing all CARs that are don't satisfy the minimum support and confidence. The checking for the classifier that infrequent rules in the incremental data are required against incremental data $\boldsymbol{d_i}$ to update their supports and confidences. In this case of the CARs which already exist in the classifier but don't satisfy minimum support and

confidence after database update will stay at the classifier because they will use for prediction phase at level II. Additionally, The CARs in the classifier and the generated frequent CARs from d_i that satisfy minsup and minconf will merge together.

All previous rules are taken a weight as follow:

1. The CARs that were exist in the classifier and generated as frequent from d_i will take very high weight.
2. The CARs which is frequent in d^+ and satisfy minsup and minconf against whole database but not exist in current classifier will take high weight.
3. The CARs that were exist in the classifier and infrequent in incremental data d^+ but still frequent in the entire database will take mid weight.
4. The CARs that were exist in the classifier but become infrequent in the whole database will take low weight.

The whole remaining CARs will rearrange according to following criteria: If there are two rules r_1 and r_2, then $r_1 > r_2$ if and only if:

1. if the weight of the r_1 is greater than the weight of r_2
2. If the weight of r_1 and r_2 is equal but the confidence of r_1 is greater than confidence of r_2.
3. If the weight and confidence of r_1 and r_2 are equal but the support of r_1 greater than the support of r_2.
4. If the weight, confidence and support are equal of r_1 and r_2, but r_1 has minimum attributes in the left-hand side than r_2.

After generating rules from d_i and merge them with classifier rules, as a result there will be a large number of rules. Accordingly there is a lot of redundant rules, which will have an impact negatively on the efficiency and accuracy of the final classifier. Thus, the rules that have been generated in previous step cannot be maintained.

The classifier of ACIM algorithm has two levels: Level I includes the rules which cover at least one case in the training dataset. Level II includes the rules that didn't cover any training data set. The rules at the level II almost have weight less than rules at level I. The main assumption of ACIM algorithm is trying to control the number of rules that exist at levels I and II (final classifier) by removing all rules that don't satisfy the minimum cover weight threshold δ.

The database coverage method is used by ACIM technique for testing all rules against whole database. The database coverage pruning technique find all the training data case that fully match by any rule, which is ranked in previous step. At the beginning, all rules cover weight is set 0 (where the cover weight is weight indicate to the number of times the rule enter to level I or level II) If at least one training case match the rule, the rule insert to the end of classifier at level I and then its cover weight (*coverw)* will be increased by 1, and all cases (tuples) in the training data will be discard. Otherwise, the rule insert into classifier at level II and then its cover weight will be decreased by 1.

3. The Framework of the Enhanced Algorithm E-ACIM

This algorithm is an enhanced version of ACIM algorithm to overcome some drawbacks such as the huge number of the generated classification rules which takes efforts to select the best ones in order to construct the classifier. Moreover, in most real world applications data usually are updated on a daily basis and once any data operations (adding, deleting, updating) are carried out on certain training data set. As a result, most of the current AC algorithms require repetitive scanning over the complete training data set and then update the results (classifier) in order to reflect changes caused by such that operations, which increases the required processing time. The main idea of this algorithm is to build a classifier when new data arrive by utilizing the old classifier without rescanning to the entire dataset as well as keep the classifier without updating if no need for that. Therefore, E-ACIM reduces the computational time for building the classifier.

Figure 1 illustrates the developed Enhanced Associative Classification based on incremental mining (**E-ACIM**) algorithm which consists of the following steps:

Step 1: Building the classifier from the training data set (***D***), then testing, and deriving the classification accuracy.

Step 2: Selecting a stratified random sample (d_{si}) from the new data which have been added to ***D*** (d_i) where this sample represents only 10%. For fair selection, the chosen 10% rows are stratified meaning the class distributions within the new added rows are considered so the selected sample contains all class labels.

Step 3: Utilizing the chosen sample (d_{si}) as a testing set for the classifier which was built in the first step, then store the accurate result.

Step 4: Comparing the accuracy which obtained in Step 3 with that of Step 1. If the accuracy of step 3 is greater than or equal to that of Step 1 then we keep the classifier produced at Step 1. Otherwise,

Step 5: builds a new classifier using associative classification based on an incremental mining algorithm **(ACIM)**[15].

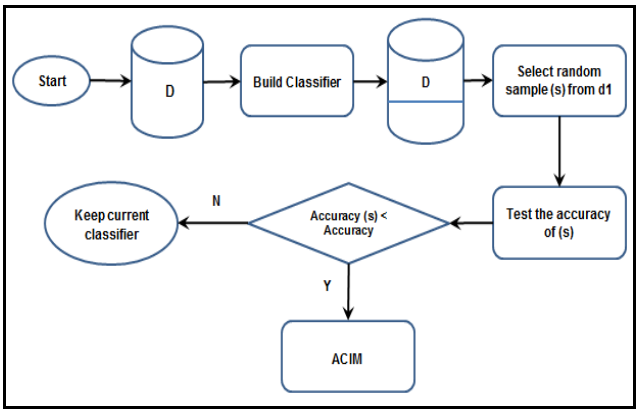

Fig. 1 E-ACIM Framework.

4. The Proposed Algorithm

In this section, we will describe the proposed algorithm which is as follows:

4.1 General Description

Table 1, presents the notations used in this section:

Table 1: Method and Notations of the Proposed Algorithm

Notation	Detail
I	Indicator of the incremental data set number
D	Original data set
d_i	Incremental data set
d_{si}	Stratified random sample from d_i
A	The accuracy of the classifier on original testing data set
a	The accuracy of classifier on d_{si}
R	Class association rule(CAR) from d_{si}
Methods	**Detail**
SRS()	Extract stratified random sample transactions from d_i
Buildclassifier_ CBA()	Build classifier using CBA algorithm and testing.
classifier_ testing()	Test the rule
ACIM()	Building the classifier using ACIM algorithm

4.2 E-ACIM Algorithm

Figure 2 shows the pseudo code of the proposed algorithm:

```
1   Inputs: Original Database (D), minsup, minconf
2   Outputs: Classifier (C)
3   // build the classifier using original data
4   buildclassifier_CBA(D)
5   A=Accuracy_of_classifier(D);
6   do until (no incremental data) {
7   //select stratified random sample
8   // from di  and use it for testing
9   //using current classifier then
10  //store  its  accuracy
11  dsi= SRS(di)
12  for each r Є dsi
13  {
14      classifier_ testing(r)
15      a = Accuracy_of_classifier (dsi)
16  }
17  // compare the accuracy of
18  // classifier on dsi with classifier
19  // accuracy on D
20  if (a < A) then
21  {
22    // build new classifier from incremental
23    // data and original data with utilizing
24   // previous classifier using classic ACIM
25   ACIM (di  U D)
26  }
27  Else
28  {
29    D=D U di
30  } } }
```

Fig. 2 The Pseudo Code of Proposed Algorithm.

According to Figure 2, in lines 4-5 the classification model is constructed using CBA algorithm from the original database (***D***), and then the accuracy is derived and stored in variable ***A***. In line 11, the algorithm selects a stratified random sample (***d_{si}***) from the incremental data set (***d_i***) where this sample represents only 10% of the incremental data, which has nontrivial size. For fair selection, the chosen 10% rows are stratified meaning the class distributions within the new added rows are considered, so the selected sample contains all class labels, then in lines 12-16 ***ds_i*** gets utilized as testing data against the CBA classifier produced in line 4. The accuracy derived from this testing is checked with that of CBA (lines 20-30). If the new accuracy is better, then the produced classifier at

line 4 or 25 is kept unchanged. Otherwise a new classifier will build from the incremental data and original data by applying **ACIM** algorithm[9]. This process is repeated whenever the training data set ***D*** gets updated (incremental data).

Thus, based on the above, the first classifier built from the original database ***D***; when increment data d_i are added to the original dataset. The random sample selected and tested using the current classifier if the accuracy and the current classifier work well enough on the incremental data; then proposed algorithm retain the existing classifier. Otherwise, (i.e. the accuracy of the current classifier on random sample is less than the previous accuracy) **ACIM** will be used to build a new classifier using the incremental data and original data ***D*** by utilizing the classifier which built in the previous phase.

5. Experimental Results

The performed experiments have been conducted to evaluate the classification accuracy and the computation time of the proposed enhanced associative classification based on incremental mining algorithm **E-ACIM** , in addition to classic **CBA** and **ACIM** algorithms.

The proposed method adopts the same processes which are used in ACIM for rule generation, pruning and prediction procedures to build the classifier from original dataset ***D***. Different data sets from the UCI machine learning repository [16] are used during the experimentation. These are "Car eval.", "led7", "page blocks", "pen digits", "waveform", "wine qu.". The selection of these data sets is based on the size and the data quality since we have looked for medium to large size training data set because we divide the data set into original and incremental data. Moreover, the proposed algorithm depends on selecting a random sample which needs large dataset, also the above data sets don't contain missing values. Explanation of the characteristics of each dataset can be found in [8]. Table 2 shows the characteristics of the using datasets.

Table 2: UCI dataset Characteristics

Data set	Number of attributes	Number of records	Class no.
Car evaluation	6	1728	4
Page block	10	5473	5
Pen digit	16	10992	10
Wav form	21	5000	3
Wine quality	12	4898	7
Led7	7	3200	10

Each UCI data set has been divided into two partitions: first one for training and another one for the incremental learning process (data to be inserted). Then, we further divided the incremental data set into five data blocks for testing purposes and particularly to perform multiple data insertions on the training data set during training and evaluation steps.

All experiments were performed on a Celeron 2.16 a computing machine with the following specifications: 1GB main memory, and running Microsoft windows XP. The LUCS-KDD implementation of the CBA algorithm is used. The main thresholds that control the number of rules generated and prediction accuracy in AC mining are minimum support and confidence for the proposed algorithm **E-ACIM**, ACIM, and CBA. These thresholds have been set to 1% and 50%, respectively. This setting is similar to other previous research conducted by AC [9, 17].

In the experiments, The first classifier built from the original training data (i.e. first part of the training data), then the accuracy of this classifier after applying TCV (Ten-Cross Validation) stores for comparison purpose. After that, new incremental data (first part of incremental data-1/5) adds to the original data. The developed algorithm selects a stratified random sample of incremental data and use it as testing data on the original classifier that obtained from the original training data then compare the achieved accuracy with the previous accuracy. If the new accuracy is less than previous accuracy the algorithm rebuilds a classifier using original and incremental data using **ACIM**, otherwise preserve the old classifier. At the end of this step the algorithm adds the incremental data to the original data and consider both as original data in the second step.

After the addition of the second part of the incremental data the previous process repeats where the stratified random sample is selected from the second part of incremental data and test on the previous classifier to decide if it will retain or build a new classifier.

The above process repeat on the five incremental training data and the average accuracy of the classifier on the five steps will be computed. If the classifier retains the accuracy of the previous step will be taken. The average runtime of the algorithm on the five phases is computed, since when the classifier retains, the cost comes from the selecting and testing stratified random sample, otherwise the cost comes from rules generation and classifier building process.

Table 3 shows the average accuracy of **CBA**, **ACIM and E-ACIM**. It is noticeable, the accuracy of proposed algorithm is better than both of CBA and ACIM over some dataset. This is a logical result due to that proposed algorithm apply the same technique that applied by ACIM. Whereas, the ACIM has two levels of rules that reduce the

using of default class and using the weight of the rule for rule ranking process, in addition to support and confidence that allow better rule to classify the unseen object. Overall, the cause of accuracy variance on the selected data sets are due to the nature of these data sets , i.e. the distribution of the class labels in the data set, number of records per class, the size of data set, and stratified random sample effects.

Table 3 shows the average accuracy after taking the average minimum cover weight threshold. Moreover, the experiments conducted on different value of the cover weight, since, it's set to {0,-1,-2,-4,-5}. The algorithm **E-ACIM**, ACIM and CBA have been applied five times. The ACIM and E-ACIM applied with different minimum cover weight threshold δ and the average of the five running of the algorithm records in table 3. The weight for very high, high, mid and low rules is set 4,3,2,1 respectively. The original training dataset and incremental dataset are maintained like the first and second part implementation of **ACIM**.

Table 3: Average Accuracy of CBA, ACIM and the proposed E-ACIM

Data set	CBA Avg. Accuracy	ACIM Avg. accuracy	E-ACIM Avg. accuracy
Car evaluation	90.90	92.1	92.1
Led7	71.90	71.62	71.55
Page blocks	93.68	92.18	92.18
Pen digits	91.30	92.42	92.41
Waveform	75.60	76.23	76.27
Wine quality	44.09	44.06	44.01

Table 4 shows the runtime of CBA and **ACIM**. The proposed algorithm needs less time for building classifier than CBA in all datasets with 2 to 7 times. This is due to that **ACIM** does not build the classifier from scratch after each data insertion. Figure 3 and Figure 4 show the average accuracy and average runtime of CBA, ACIM and **E-ACIM** graphically.

Table 4: Average running time for CBA, ACIM and the proposed E-ACIM

Data set	CBA Avg. runtime in sec.	ACIM Avg. runtime in sec.	E-ACIM Avg. runtime in sec.
Car evaluation	0.26	0.21	0.26
Led7	0.36	0.23	0.18
Page blocks	1.15	0.85	0.52
Pen digits	145.0	35.16	21.56
Waveform	140.4	45.21	32.43
Wine quality	3.55	2.01	1.61

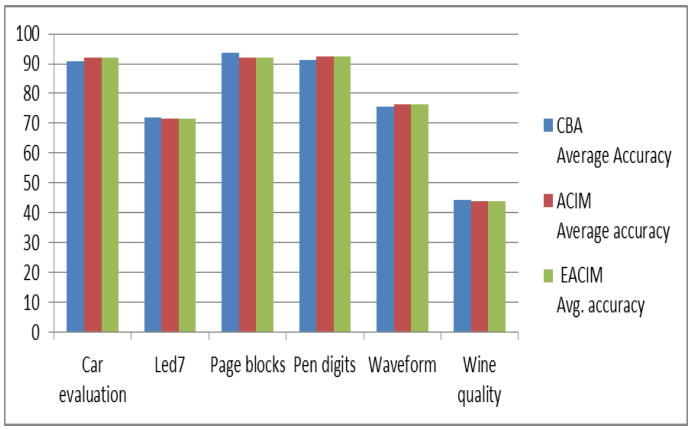

Fig. 3 Average Accuracy of CBA, ACIM and the proposed E-ACIM.

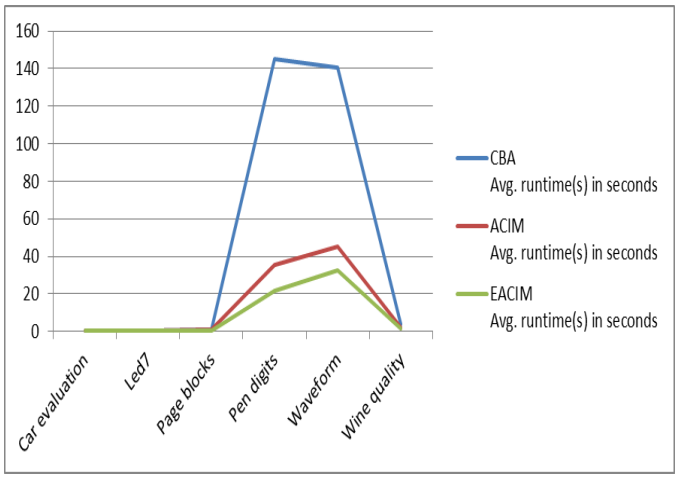

Fig. 4 Average Runtime of CBA, ACIM and the proposed E-ACIM.

6. Conclusions

This paper introduced an enhanced associative classification based on an incremental mining algorithm called **E-ACIM**. This algorithm deals with the data insertion problem in associative classification context. The method uses the stratified random sample of incremental data as a testing data to evaluate the classifier derived from the original training data set. The classifier is constructed again using **ACIM** if the accuracy of the classifier on the testing data sample is less than that of the classifier derived from the original data. The experimental results against six UCI data collection showed that, **E-ACIM** reduces the computational time for five data sets comparing with **ACIM** and classic **CBA**. Also the suggested algorithm almost provided the same level of accuracy for both the **ACIM** and **CBA**.

References

[1] Fayyad U., Piatetsky-Shapiro G., Smith G. and Uthurusamy R. Advances in knowledge discovery and data mining, . AAAI Press 1998.
[2] Toshi C. and Neelabh S., "Incremental Mining on Association Rules", Research Inventy: International Journal of Engineering and Science, Vol. 1, No. 11, 2012, pp. 31-33.
[3] Nath B., Bhattacharyya D. K., and GhoshA., "Incremental association rule mining: a survey," WIREs Data Mining KnowlDiscov, vol. 3, No. 3, 2013, pp. 157–169.
[4] Cheung D., Han J., Ng V. and Wong C., "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique", in the International Conference on Data Engineering, 1996, pp.106-114.
[5] Cheung D., Lee S. and Kao B., "A general Incremental Technique for Mining Discovered Association Rules", in International Conference on Database System for Advanced Applications, 1997, pp. 185-194.
[6] Tsai P., Lee C. and Chen A., "An efficient approach for incremental association rule mining". In the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining, 1999, pp. 74-83.
[7] Petko V., Rokia M., Mohamed R., Hacene Robert G., " Incremental Maintenance of Association Rule Bases", in Proceedings of the 2nd Int. Workshop on Data Mining and Discrete Mathematics, 2003, San Francisco (CA/US).
[8] Chang C., Li Y. and Lee J., "An Efficient Algorithm for Incremental Mining of Association Rules", in proceeding of the 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications, 2005.
[9] Liu B., Hsu W. and Ma Y., "Integrating classification and association rule mining", in proceedings of the KDD, 1998, pp. 80-86, New York, NY.
[10] Quinlan J. and Cameron-Jones R., "FOIL: A midterm report", in proceedings of the European Conference on Machine Learning, 1993, pp. 3-20, Vienna, Austria.
[11] Thabtah F., Cowling P. and Peng Y., "MCAR: Multi-class classification based on association rule approach", in proceeding of the 3rd IEEE International Conference on Computer Systems and Applications, 2005, pp. 1-7, Cairo, Egypt.
[12] Yin X. and Han J., "CPAR: Classification based on predictive association rule". In proceedings of the SDM, 2003, pp. 369-376, San Francisco, CA.
[13] Antonie M., Zaïane O. R. and Coman A., Associative Classifiers for Medical Images, Lecture Notes in Artificial Intelligence 2797, Mining Multimedia and Complex Data, 2003, pp. 68-83, Springer-Verlag.
[14] Yoon Y. and Lee G., "Practical application of associative classifier for document classification", in proceeding of Asia Information Retrieval Symposium, 2005, pp. 467-478, Jeju-island, Korea.
[15] Nababteh M., Fayoumi M., Aljuma A. and Ababneh J., "Associative Classification Based On Incremental Mining (ACIM)", International Journal of Computer Theory and Engineering (IJCTE), Vol.6, No. 2, 2014.
[16] Merz C. and Murphy P., UCI repository of machine learning databases. Irvine, CA, University of California, Department of Information and Computer Science, 1996.
[17] Li W., Han J., and Pei J., "CMAR: Accurate and efficient classification based on multiple-class association rule", in proceedings of the ICDM'01, 2001, pp. 369-376. San Jose, CA.

Mustafa A. Al-Fayoumi received the B.S. degree in computer science from Yarmouk University, Irbid, Jordan, in 1988. He received the M.S. degree in computer science from the University of Jordan, Amman, Jordan, in 2003. In 2009, he received a Ph.D. degree in computer science from the Faculty of Science and Technology at Anglia University, UK. In 2009, he joined the Al-Zaytoonah University, in Jordan, as an assistant professor. Currently, he is assistant professor and chairman of computer science department at Salman bin Abdulaziz University, Saudi Arabia. His research interests include areas like computer security, cryptography, identification and authentication, wireless and mobile networks security, e-application security, simulation and modeling, algorithm analyzes and design, information retrieval, data mining and any other topics related to them.