Mining Student Evolution Using Associative Classification and Clustering

Kifaya S. Qaddoum, Faculty of Information, Technology Philadelphia University
Amman, Jordan, E-mail: kqaddoum@philadelphia.edu.jo

Abstract

Associative classification (AC) is a branch in data mining that utilises association rule discovery methods in classification problems.This paper idea aims to discuss and evaluate a modeling approach for student evolution. It is developed as a component of an adaptive achievement system. At the beginning of the process, associations of student achievement results are found based on each student's factors that affect learning process, which finds the relationship between student evolution during years of study and understanding the modular scheme, that finds the main effect of enrolling on the correct modules i.e. getting the right advice and student support regarding choosing modules, meeting all the necessary prerequisites, having summer courses, and taking in consideration the student's high school grades, as well as finding the relationship between modules type and student gender . Clustering [10], or unsupervised classification, method is employed to model this task.

The goal of clustering is [7] to objectively partition data into homogeneous groups such that the within group object similarity and the between group object dissimilarity are determined. Clustering here is used to model student achievement according to predefined criterion functions that measure similarity among students who grant certain goal having the same conditions using data collected from University Database. A clustering method is developed for this step. We evaluated the student progress according to associations between different factors using data collected. We concluded the performance of those groups using these two approaches.

Now, the need for solid information about student evolution and how to improve it has only grown in importance for state policy. The compelling metaphor of increasing flow through the "educational pipeline" is now common in state policy discussions, fueled by more vocal recognition by business and civic leaders of the importance of the critical "supply chain" of educational capital in their states.

Keywords
Associative Classification, Classification, Itemset, Clustering.

1. Introduction

Information system is developing very rapidly in data warehousing. Due to the diversity of data sets, efficient retrieval of information is very important for decision making. Data mining is the science of extracting meaningful information from these large data warehouses [18]. Data mining and knowledge discovery techniques have been applied to several areas including, market analysis, industrial retail, decision support and financial analysis. Knowledge Discovery from Databases (KDD) [6] involves data mining as one of its main phases to discover useful patterns. Other phases in KDD are data selection, data cleansing, data reduction, pattern evaluation and visualisation of discovered information [4].

Since it has been introduced, Association Rule Mining (ARM) [1] has received a great deal of attention by researchers and practitioners among data mining. ARM is an undirected or unsupervised data mining technique, which works on variable length data, and it produces clear and understandable results. It has a simple problem statement, that is, to discover relationships or correlations in a set of items and consequently find the set of all subsets of items or attributes that frequently occur in many database records or examples, and additionally, to extract the rules telling us how a subset of items influences the presence of another subset.

2. Association Rule Mining

The association mining task simply can be stated as follows [1]: Let I be a set of items, and D a database of examples, where each example has a unique identifier (*tid*) and contains a set of items. A set of items is also called an *itemset*. An *itemset* with k items is called a *k-itemset*. The *support* of an *itemset* X, denoted $\sigma(X)$, is the number of examples in D where it occurs as a subset. An *itemset* is *frequent* or *large* if its support is more than a user-specified *minimum support (min sup)* value.

An *association rule* is an expression $A \Rightarrow B$, where A and B are *itemsets*. The support of the rule is the joint probability of an example containing both A and B, and is given as $\sigma\ (A \cup B)$. The *confidence* of the rule is the conditional probability that an example contains B, given that it contains A, and is given as $\sigma\ (A \cup B) / \sigma\ (A)$. A rule is *frequent* if its support is greater than *min sup*, and it is *strong* if its confidence is more than a user-specified *minimum confidence (min conf)*.

3. Problem Definition

The main objective of data mining is to find interesting/useful knowledge for the user, as Rules are

an important form of knowledge; some existing research has produced many algorithms for rule mining. These techniques use the whole dataset to mine rules and then filter and/or rank the discovered rules in various ways to help the user identify useful ones.

There are many potential application areas for association rule technology which include catalog design, store layout, customer segmentation, telecommunication alarm diagnosis, and so on.

The data mining task is to generate all association rules in the database, which have a support greater than *min sup*, i.e., the rules are frequent, and which also have confidence greater than *min conf*, i.e., the rules are strong. Here we are interested in rules with a specific item, called the *class*, as a consequent, i.e., we mine rules of the form $A \Rightarrow c_i$ where c_i is a class attribute ($1 \leq i \leq k$).

This task can be broken into two steps:

1. Find all frequent *itemsets* [17] having minimum support for at least one class c_i. The search space for enumeration of all frequent *itemsets* is 2^m, which is exponential in *m*, the number of items.

2. Generate strong rules having minimum confidence, from the frequent *itemsets*. We generate and test the confidence of all rules of the form $X \Rightarrow c_i$, where X is frequent. For example, consider the sales database of a bookstore [20] shown in Figure 1, where the objects represent customers and the attributes represent books. The discovered patterns are the set of books most frequently bought together by the customers. An example could be that, "40 percent of the people who buy Jane Austen's *Pride and Prejudice* also *buy Sense and Sensibility*". The store can use this knowledge for promotions, shelf placement, etc.

DISTINCT DATABASE ITEMS

Jane Austen	Agatha Christie	Sir Arthur Conan Doyle	Mark Twain	P. G. Wodehouse
A	C	D	T	W

DATABASE

Transcation	Items
1	A C T W
2	C D W
3	A C T W
4	A C D W
5	A C D T W
6	C D T

ALL FREQUENT ITEMSETS

MINIMUM SUPPORT = 50%

Support	Itemsets
100% (6)	C
83% (5)	W, CW
67% (4)	A, D, T, AC, AW CD, CT, ACW
50% (3)	AT, DW, TW, ACT, ATW CDW, CTW, ACTW

Fig 2 Distinct Database items

There are five different items (names of authors the bookstore carries), i.e., $I = \{A, C, D, T, W\}$, and the database consists of six customers who bought books by these authors. Figure1 [12] shows all the frequent *itemsets* that are contained in at least three customer transactions, i.e., *min sup* =50 percent.

There is one main difference between classification [3] and ARM which is the outcome of the rules generated. In case of classification, the outcome is pre-determined, i.e. the class attribute. Classification also tends to discover only a small set of rules in order to build a model (classifier), which is then used to forecast the class labels of previously unseen data sets as accurately as possible. On the other hand, the main goal of ARM is to discover correlations between items in a transactional data set. In other words, the search for rules in classification is directed to the class attribute, whereas, the search for association rules are not directed to any specific attribute.

Associative Classification (AC) is a branch in data mining that combine's classification and association rule mining. In other words, it utilises association rule discovery methods in classification data sets. Many AC algorithms have been proposed in the last few years, i.e. [13], [14], [16], and produced highly competitive results with respect to classification accuracy if compared with that of traditional classification approaches such as decision trees , probabilistic [3] and rule induction.

4. Associative Classification Problem and Related Works

According to [16] the AC problem was defined as: Let a training data set T has m distinct attributes A1, A2, … , Am and C is a list of class labels. The number of rows in T is denoted |T|. Attributes could be categorical (meaning they take a value from a finite set of possible values) or continuous (where they are real or integer). In the case of categorical attributes, all possible values are mapped to a set of positive integers. For continuous attributes, a discretisation method is first used to transform these attributes into categorical ones.

Definition 1: An *item* can be described as an attribute name A_i and its value a_i, denoted (A_i, a_i).

Definition 2: The j_{th} *row* or a *training object* in *T* can be described as a list of items $(A_{j1}, a_{j1}), \ldots, (A_{jk}, a_{jk})$, plus a class denoted by c_j.

Definition 3: An *itemset* can be described as a set of disjoint attribute values contained in a training object, denoted $< (A_{i1}, a_{i1}), \ldots, (A_{ik}, a_{ik})>$.

Definition 4: A *ruleitem r* is of the form $<cond, c>$, where condition *cond* is an itemset and $c \varepsilon C$ is a class.

Definition 5: The actual occurrence (*actoccr*) of a *ruleitem r in T* is the number of rows in *T* that match *r's* itemset.

Definition 6: The support count (*suppcount*) of *ruleitem r* = <*cond*, *c*> is the number of rows in *T* that matches *r's* itemset, and belongs to a class *c*.
Definition 7: The occurrence (*occitm*) of an itemset *I* in *T* is the number of rows in *T* that match *I*.
Definition 8: An itemset *i* passes the minimum support (*minsupp*) threshold if (*occitm*(*i*)/|*T*|) ≥ *minsupp*. Such an itemset is called *frequent* itemset.
Definition 9: A *ruleitem r* passes the *minsupp* threshold if, *suppcount*(*r*)/ |*T*| ≥ *minsupp*. Such a *ruleitem* is said to be a *frequent ruleitem*.
Definition 10: A *ruleitem r* passes the minimum confidence (*minconf*) threshold if *suppcount*(*r*) / *actoccr*(*r*) ≥ *minconf*.
Definition 11: An associative rule is represented in the form: $cond \rightarrow c$, where the antecedent is an itemset and the consequent is a class.

The problem of AC [2] is to discover a subset of rules with significant supports and high confidences. This subset is then used to build an automated classifier that could be used to predict the classes of previously unseen data. It should be noted that MinSupp and MinConf terms in ARM are different than those defined in AC since classes are not considered in ARM, only itemsets occurrences are used for the computation of support and confidence.

Classification Based on Associations (CBA) was presented by [13] and it uses Apriori candidate generation method [1] for the rule discovery step. CBA operates in three steps, where in step 1, it discretises continuous attributes before mining starts. In step 2, all frequent ruleitems which pass the MinSupp threshold are found, finally a subset of these that have high confidence are chosen to form the classifier in step3. Due to a problem of generating many rules for the dominant classes or few and sometime no rules for the minority classes, CBA (2) was introduced by [12], which uses multiple support thresholds for each class based on class frequency in the training data set. Experiment results have shown that CBA (2) outperforms CBA and C4.5 in terms of accuracy.

Classification based on Multiple Association Rules (CMAR) adopts the FP-growth ARM algorithm [11] for discovering the rules and constructs an FP-tree to mine large databases efficiently [14]. It consists of two phases, rule generation and classification. It adopts a FP- growth algorithm to scan the training data to find the complete set of rules that meet certain support and confidence thresholds. The frequent attributes found in the first scan are sorted in a descending order, i.e. F-list. Then it scans the training data set again to construct an FP-tree. For each tuple in the training data set, attribute values appearing in the F-list are extracted and sorted according to their ordering in the F-list. Experimental results have shown that CMAR is more accurate than CBA and C4.5 algorithms. The main drawback documented in CMAR is the need of large memory resources for its training phase.

Classification based on Predictive Association Rules (CPAR) is a greedy method proposed by [9]. The algorithm inherits the basic idea of FOIL in rule generation [15] and integrates it with the features of AC. Multi-class Classification based on Association Rule (MCAR) is the first AC algorithm that has used a vertical mining layout approach [20] for finding rules. As it uses vertical layout, the rule discovery method is achieved through simple intersections of the itemsets Tid-lists, where a Tid-list contains the item's transaction identification numbers rather than their actual values. The MCAR algorithm consists of two main phases: rules generation and a classifier builder. In the first phase, the training data set is scanned once to discover the potential rules of size one, and then MCAR intersects the potential rules Tid-lists of size one to find potential rules of size two and so forth. In the second phase, the rules created are used to build a classifier by considering their effectiveness on the training data set. Potential rules that cover a certain number of training objects will be kept in the final classifier. Experimental results have shown that MCAR achieves 2-4% higher accuracy than C4.5, and CBA.

Multi-class, Multi-label Associative Classification (MMAC) algorithm [16] consists of three steps: rules generation, recursive learning and classification. It passes over the training data set in the first step to discover and generate a complete set of rules. Training instances that are associated with the produced rules are discarded. In the second step, MMAC proceeds to discover more rules that pass MinSupp and MinConf from the remaining unclassified instances, until no further potential rules can be found. Finally, rule sets derived during each iteration are merged to form a multi-label classifier that is then evaluated against test data. The distinguishing feature of MMAC is its ability to generate rules with multiple classes from data sets where each data objects is associated with just a single class. This provides decision makers with useful knowledge discarded by other current AC algorithms.

To the best of the authors' knowledge and during the learning step, most of the above AC algorithms join frequent itemsets of size K regardless of their class values to derive candidate itemsets of size K+1. Whereas, our proposed training algorithm only joins frequent itemsets with common class values of size K to produce candidate itemsets of size K+1. This significantly reduces costs associated with memory usage and training time as discussed in details in Section 4.

5. Clustering

Clustering which considered as the most important *unsupervised learning* problem [10], [8], [7] ; so, as every other problem of this kind, it deals with finding a *structure* in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A *cluster* is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how

to decide what constitutes a good clustering? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs. In our case, we are interested in finding representatives for homogeneous groups (*data reduction*), in finding "natural clusters" and describe their unknown properties (*"natural" data types*), in finding useful and suitable groupings (*"useful" data classes*) or in finding unusual data objects (*outlier detection*), for students groups according to their evolution during the specified time of study.

6. Association Rule And Clustering Algorithm For Modeling Student Evolution

This proposed Algorithm is an iterative algorithm that counts *itemsets* of a specific length in a given database pass. The process starts by scanning all transactions in the database and computing the frequent items. Next, a set of potentially frequent candidate 2-*itemsets* is formed from the frequent items. Another database scan is made to obtain their supports. The frequent 2-itemsets are retained for the next pass and the process is repeated until all frequent *itemsets* have been enumerated.

There are three main steps in the algorithm:

1. Generate candidates of length k from the frequent (k-1) length *itemsets*, by a self join on F_{k-1}. For example, If

   F_2= {*AB, AC, AD, AE, BC, BD, BE*}.
   Then we find that :

   C3 = {ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE}.

2. Prune any candidate with at least one infrequent subset. As an example, ACD will be pruned since CD is not frequent. After pruning, we get a new set C3 = {ABC, ABD, ABE}.
3. Scan all transactions to obtain candidate supports. The candidates are stored for support counting.

Example Let L3 be {{1 2 3}, {1 2 4}, {1 3 4}, {13 5}, {2 3 4}}. After the join step, C4 will be {{1 2 3 4}, {1 3 4 5}}. The prune step will delete the *itemset* {1 3 4 5} because the *itemset* {1 4 5} is not in L3. We will then be left with only {1 2 3 4} in C4.

Data Partition: 70% Training, and 30% Validation, since Models are constructed using training data sets and evaluate model performance using validation data sets, and using other data sources as testing data sets.

We used F1 evaluation measure as the base of our comparison, where F1 [19] is computed based on the following equation:

$$F1 = \frac{2 * \Pr ecision * \mathrm{Re} call}{\mathrm{Re} call + \Pr ecision}$$

Precision and recall are widely used evaluation measures in IR and ML, where according to Table 2,

$$\Pr ecision = \frac{X}{(X+Y)}$$

$$\mathrm{Re} call = \frac{X}{(X+Z)}$$

Table 2 : Data possible sets based on a query in IR

Iteration	Relevant	Irrelevant
Data Retrieved	X	Y
Data not Retrieved	Z	W

To explain precision and recall, let's say someone has 5 blue and 7 red tickets in a set and he submitted a query to retrieve the blue ones. If he retrieves 6 tickets

where 4 of them are blue and 2 that are red, it means that he got 4 out of 5 blue (1 false negative) and 2 red (2 false positives). Based on these results, precision=4/6 (4 blue out of 6 retrieved tickets), and recall= 4/5 (4 blue out of 5 in the initial set).

For objectively partitioning data into homogeneous groups, it is necessary to define criterion functions that measure similarity among objects. Various criterion functions and methodologies have been developed for temporal data clustering systems. They can be grouped into three main categories: (i) proximity based methods, (ii) feature based methods, and (iii) model-based methods.

7. Experimental Analysis

7.1 Data Description

I collected and stored all student activities in the database. Data collected from Computer Science I (CS-I) students in 2005 was used for this experiment. The collected data contains information from 166 students. Depending on a student's performance and the type of student identified by its learned model. The analysis done on this students data was through periods and semesters that student spent and the grades he obtained during each semester, where the increase or decrease on his grades leads to modification on his predicted evolution for the coming semesters.

7.2 Experimental Design

The first experiment compared the student models generated using the classification approach on the static survey data, and those using the clustering approach on the temporal student online data. The

classification model learned from the CS-1 students in 2004 -as shown in Figure 2- was applied to students in Spring 2005 after each answered the six learning behavior related questions. Each student was classified into one of three learning categories: Reinforcement type(A), Challenging type(B), and Regular type(C). For the same group of students, the Markov chain based clustering was applied to the temporal lab data deriving a set of classes corresponding to the set of student learning models. Manually analyzing

Level	year	G1	G2	GPA1	GPA2
3	2000	9	6	67.5	67.5
1	2001	15	9	54.2	58
2	2001	15	6	47	53.4
1	2002	15	15	60.4	61.2
2	2002	18	6	46	56.4
1	2003	0		0	56.4
2	2003	0	0	0	56.4
1	2004	15	12	69.5	61.9
2	2004	15	12	52.2	60
3	2004	9	9	62.7	61.2
1	2005	18	18	62.2	62.1
2	2005	15	12	46.8	60.3
1	2006	12	12	53	60.2
1	2001	15	15	60	60
2	2001	12	9	60	60
1	2002	15	3	67	60.8
2	2002	12	12	64.8	62
1	2003	0	0	0	62
2	2003	0	0	0	62
1	2004	12	12	71.3	64.2
2	2004	12	12	63.3	64
1	2005	12	12	62.8	63.8
2	2005	12	12	59.8	63.2

Fig 2. sample of students data sets

these models leads to a labeling of learning types for these clusters.

In order to compare whether the student categorization derived from the two approaches resemble each other, we compared the category labels assigned to the students. To determine which approach gives a better categorization of the students, I objectively measured the quality of the models derived in terms of the between cluster dis-similarity and within cluster dis-similarity. The derived student learning models are considered of better quality if the models representing different categories are as unique, or as dis-similar to each other as possible. In addition, the student models are considered better quality if students presented by each category are homogeneous in learning style than if there are subgroups following significantly different learning style.

This proves that, after the first level cluster, the students categorized into the same group share very similar behavior pattern/model. They could not be further split into different groups, as in the case of cluster "C2" (distance value 0.0 is put in the table entry), or only relatively similar models could be derived. In the case of the classification approach, since the first level classification did not successfully partition students into homogeneous groups based on their data.

8. Conclusion

This paper showed that using Associative Classification and Clustering was effective in finding relations and associations between students raising among given categories. We evaluated the student progress according to associations between different factors using data collected. We concluded the performance of those groups using these two approaches, where we can mine the expected groups for each student. For future work this study should use different categorization algorithms which handle a dynamic and updated data for the students.

9. References

[1]Agrawal, R. and Srikant, R. 1994. "Fast algorithms for mining association rules." VLDB-94, 1994.

[2]B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In Proc. of 4th Intl. Conf. on Knowledge Discovery and Data Mining (KDD), Aug. 1998.

[3]Duda, R., and Hart, P. (1973) Pattern classification and scene analysis. John Wiley & son, 1973.

[4]Elmasri and Navathe, Fundamentals of Database Systems (5th Edition) 2006.

[5]Elmasri, R., Navathe, S. (2007) Fundamentals of database systems, Fourth Edition, Addison-Wesley.

[6]Fayyad, U., Piatetsky-Shapiro, G., Smith, G., and Uthurusamy, R. (1998) Advances in knowledge discovery and data mining. AAAI Press, 1998.

[7]Fisher, D., Data Mining Tasks and Methods: Clustering: Conceptual Clustering, Handbook of Data Mining and Knowledge Discovery, 388-396, 2002.

[8]Gobert, J., & Buckley, B. C., & Horwitz, P. (April, 2006). Technology-enabled assessment of model-based

learning and inquiry skills among high school biology and physics students. To be presented at the American

[9]Han, J., Pei, J., and Yin, Y. (2000) Mining frequent patterns without candidate generation. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, (pp. 1-12). Dallas, Texas.

[10]J.A. Fernández Pierna, and D.L. Massart, "Improved algorithm for clustering tendency", Anal. Chim. Acta, Vol 408, pp. 13–20, 2000.

[11]Li, W., Han, J., and Pei, J. (2001) CMAR: Accurate and efficient classification based on multiple-class association rule. Proceedings of the ICDM'01 (pp. 369-376). San Jose, CA.

[12]Liu, B., Hsu, W., and Ma, Y. (1999) Mining association rules with multiple minimum supports. Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (pp.337-341). San Diego, California.

[13]Liu, B., Hsu, W., and Ma, Y. (1998) Integrating classification and association rule mining. Proceedings of the KDD, (pp. 80-86). New York, NY.

[14]Li, C., A Bayesian Approach to Temporal Data Clustering using the Hidden Markov Model Methodology, PhD thesis, Vanderbilt University, December 2000.

[15]Park, Calif.: AAAI Press, 1996.

[16]Thabtah, F., Cowling, P., and Peng, Y. (2004) MMAC: A new multi-class, multi-label associative classification approach. Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM '04), (pp. 217-224). Brighton, UK. (*Nominated for the Best paper award*).

[17]Toivonen, and A. Inkeri Verkamo, "Fast *Discovery of Association Rules", Advances in Knowledge Discovery and Data Mining*", U. Fayyad and et al., eds., pp. 307±328, Menlo

[18]Witten, I., and Frank, E. (2000) Data mining: practical machine learning tools and techniques with Java implementations. San Francisco: Morgan Kaufmann.

[19]Van Rijsbergan C. J., Information Retrieval, Butterworths, 1979.

[20] Zaki, M., Parthasarathy, S., Ogihara, M., and Li, W. (1997) New algorithms for fast discovery of association rules. *Proceedings of the 3rd KDD Conference* (pp. 283-286). Menlo Park, CA.