Mining Association Rules: A Database Perspective

Dr. Abdallah Alashqur

Faculty of Information Technology
Applied Science University
Shafa Badran, Amman, JORDAN

Summary

Mining of association rules in a relational database is important because it discovers new knowledge in the form of association rules among attribute values. This enables business managers to make the right decisions pertaining to their businesses. However, association rule mining concepts and algorithms have been traditionally applied to a specific representation of data called *market basket data* representation or *transactional* representation. Data in a relational database is not represented in a market basket data format but rather in a format that adheres to the relational data model. Data in a relational database has to be converted to the market basket data representation before data mining algorithms can be applied to it. This requires the application of tedious conversion processes on large quantities of data before such algorithms can be applied. In this paper, we describe how association rules can be expressed directly in the context of the relational data model, and not based on market basket data representation. Many of the association rule mining concepts are re-defined in this paper in a way that makes them conform to the relational model of data.

Keywords:
Data Mining, Association Rule, Itemset, Relational Model, Relational Database.

1. Introduction

Data Mining, which some times is referred to as knowledge discovery in databases, aims at finding patterns, trends, and correlations among data items in large databases. The patterns or trends that data mining discovers are usually of benefit to the business and guides management in the decision making process. To briefly demonstrate how business can benefit from data mining, assume a database that contains data pertaining to members of a gym club. Information about how long a member maintained his/her valid membership is stored in the database. In addition, information regarding how a member was originally introduced to the club (e.g., via referral, ad in a news paper) is also stored. By mining this database, a pattern may be discovered, where members who were introduced by referral tend to maintain their membership for longer periods, on average, than ones who came to know about the club through newspapers. Business managers may make use of this discovered knowledge by offering a discount to existing members if they persuade a friend or a relative to join the club.

Association rules mining [1-8] finds items in the database that are likely to be associated together. If one item appears in a record, there is a high probability the associated item also appears in the same record. Historically, research in the area of mining association rules in relational databases focused, among other things, on introducing efficient algorithms for finding frequent patterns and association rules. Examples include the Apriori algorithm and its variations and enhancements [5,6], the FP-Growth algorithm [8] , and GenMax [3].

Traditionally, in data mining literature, algorithms for mining association rules have been applied to the typical *market basket* data representation, where data is represented in a transactional form [9]. In this representation, each record has a set of items that are purchased together in one transaction. Table 1 shows an example of such representation.

Table 1: Market Basket Data format

TRANSACTION_ID	PURCHASED_ITEMS
T1	Milk, Meat, Pepsi
T2	Meat
T3	Milk, Coffee, Pepsi

However, in conventional relational databases, data is not represented in a market basket data format since multi-valued attributes are not allowed by the database normalization process. Transforming huge tables to this format, every time one needs to apply mining algorithms to them, is very inefficient. Further, many database tables cannot be easily represented in a market basket data format because the data in those relations does not lend itself easily to that representation. This hinders the practical use of these mining algorithms and limits their portability.

In this paper, we use the relational database representation of normalized relations as defined by the relational data model [10,11] as a basis to describe association rules and to compute their *support* and

Manuscript received December 5, 2008
Manuscript revised December 20, 2008

confidence. In Section 2, we present a database relation, GYM_EX_MEMBERS, that is used in the rest of the paper to perform the association rules computations in order to demonstrate the ideas of this paper. In Section 3, we define itemsets in the context of the relational data model and introduce *itemset intention*, which is used as a template shared by multiple itemsets. In Section 4, we define association rules in the context of the relational model and introduce *association rule intention*, which is used as a template shared by multiple association rules. Section 5 shows how association rules relating attributes from multiple tables can be mined. Finally, some conclusions are presented in Section 6.

2. A Relational Database Table

In this section, we briefly describe a database relation (a relation in database terminology refers to a table in the database), that is used in the remainder of this paper to describe some of the association rule mining concepts.

The relation, as shown in Table2, contains data pertaining to ex-members of a gym club, which represents the data that is kept in the database for members who terminate their membership. This data includes AGE, GENDER, MEMBERSHIP_DURATION (how long a member maintained a valid membership in the club), HOME_DISTANCE (how far a member's residence is from the club location), and HOW_INTRODUCED (how a member was originally introduced to the club such as by referral or by seeing an advertisement in a newspaper). In other words, the schema of the relation is:

GYM_EX_MEMBERS (ID, AGE, GENDER,
MEMBERSHIP_DURATION,
HOME_DISTANCE,
HOW_INTRODUCED)

Further, we assume that the data in the relation has been discretized (or categorized). Discretization of data is a popular pre-processing technique in data mining and is described in [7]. Table2 shows this relation as populated with sample data. In real life situations, a large club, with many branches, may have millions of ex-members, thus millions of tuples (a tuple in the relational database terminology refers to a record or a row in the table).

Association rule mining in a relational database discovers frequent data patterns and association rules that are useful for business. For example, by mining the data in Table 2, an association pattern between MEMBERSHIP_DURATION and HOME_DISTANCE can be discovered. According to the given data, most members who live close to the club tend to maintain their membership for a longer period than those who live far from the club location. The discovery of this pattern can be beneficial to business since the club manager can then launch a membership drive by going door-to-door to convince residents in the club neighborhood to join the club.

Table 2: GYM_EX_MEMBERS

ID	AGE	GENDER	MEMBERSHIP DURATION	HOME_ DISTANCE	HOW_ INTRODUCED
1	young	f	long	close	news_paper
2	middle	m	short	far	news_paper
3	senior	f	long	close	referral
4	senior	f	long	close	referral
5	young	f	long	far	news_paper
6	middle	m	short	close	news_paper
7	senior	m	short	far	news-paper
8	senior	f	long	close	referral
9	young	f	long	close	referral
10	middle	f	long	far	news_paper
11	middle	m	short	far	news_paper
12	senior	f	long	close	referral
13	senior	m	short	far	referral

It is not straight forward to transform the data shown in Table 2 to market basket data representation especially if the quantity of data is very large. On the other hand, data represented in a market basket data format can be converted to the relational representation. For example, the data in Table 1 can be converted to a relational representation by making a column for each item and indicating "purchase" or "no-purchase" by means of a binary representation ("1" or "0"). In other words, the relation of Table 1 is transformed to the equivalent binary representation shown in Table 3.

Table 3: Binary representation

Tras_ID	Milk	Coffee	Meat	Pepsi
T1	1	0	1	1
T2	0	0	1	0
T3	1	1	0	1

3. Itemsets and Itemset Intentions

In this section, we describe what an itemset means. We also introduce the definition of *itemset intention* as a template shared by multiple itemsets.

3.1 Itemsets

Itemsets have been defined in data mining literature, but in the context of market basket data and not based on the relational data model. Below, we recast this definition in the context of the relational model of data. An itemset in this context can be defined as *a set of items such that no two items belong to the same attribute (i.e, no two items are drawn from the same attribute domain, where a domain represents the set of valid values defined for an attribute)*. For example, {m, short, far} is a valid itemset (IS) while {m, short, far, close} is not a valid IS since 'far' and 'close' are two items that belong to the same attribute, which is HOME_DISTANCE. Stated formally, the following is the definition of a valid itemset.

{I_1, I_2,.. I_n} is valid IS iff

$$(\neg \exists I_j)\ (\neg \exists I_k)\ \ (j \neq k \wedge (Attr\ (I_j) = Attr\ (I_k))$$

Where *I* is an item from the relation (i.e. an attribute value) and *Attr (I)* is a function that returns the attribute name of item *I*. Logical AND is represented by "∧".

In Table 2, the domains of attributes are assumed, for simplicity, to be mutually exclusive. If these domains are not mutually exclusive, then one must qualify attribute values by their attribute names. Therefore, in this case, the itemset {short, news_paper} needs to be written as {MEMBERSHIP_DURATION.short, HOW_INTRODUCED.news_paper}. Note that, for clarity, we use upper case letters for attribute names and lower case letters for attribute values. An itemset that contains k items is referred to as *k-itemset*.

The interestingness of an itemset is measured by the percentage of tuples in the relation that contain the itemset. This measure is referred to, in data mining literature, as *support*. In other words, the support is the probability *P* that the itemset exists in the table.

$$Support\,(itemset) = P(itemset) = \frac{Num\ of\ tuples\ containing\ itemset}{Total\ Number\ of\ tuples} \times 100$$

As an example, based on the state shown in Table 2, the support of the 1-itemset {close} = (7/13) X 100 = 54%. The support of the 3-itemset {young, f, referral} = (1/13) X 100 = 7.7%. The support could be zero in case the itemset does not exist at all in the relation, such as {m, long} which has a support of zero. Normally, the user of a data mining tool supplies the minimum support *minsup* of interest. The data mining tool then finds the itemsets whose support is equal to or greater than *minsup*. Itemsets that satisfy the minimum support are referred to as *frequent* itemsets. For example, if the specified *minsup* value is 50%, then {young, f, referral} and {m, long} are not a frequent itemsets, while {close} is a frequent itemset.

3.2 Itemset intention

Following the terminology of the relational data model, we introduce the definition of *itemset intention* (ISI). Such definition does not exist in the context of *market basket* data representation. An *itemset intension* (ISI) is a subset of the attributes of a relation. For example, in Table 2,
{HOME_DISTANCE, MEMBERSHIP_DURATION} is an ISI. The itemsets that consist of actual attribute values from these two attributes are instantiations of this itemset intension and we refer to them as *itemset extensions* or simply *itemsets*, as defined in Section 3.1 above. In Table 2, the itemsets that are instantiations of this ISI {HOME_DISTANCE, MEMBERSHIP_DURATION} are as shown below.

{close, long}, {far, long},
{close, short}, {far, short}.

An itemset *IS* is said to be an instantiation of an itemset intention *ISI* if the cardinality of *IS* is the same as the cardinality of *ISI* and each item in *IS* is drawn from an attribute (i.e., a column) in *ISI*. Let the symbol "$\sqsubset$" *denote* "instantiation of" and let CAR (S) be a function that returns the cardinality of set S. We formally define the relationship between an itemset and its itemset intention as follows.

IS $\sqsubset$ ISI iff CAR (IS) = CAR (ISI) AND
($\forall I_j \in$ IS) (Attr (I_j) $\in$ ISI)

"I" is an item in the itemset IS and Attr (I) returns the attribute name of item I. Note that the formal definition of itemset, as introduced in Section 3.1, prevents any two values in the itemset from belonging to the same attribute.

4. Association Rules and Association Rule Intensions

In this section, we first define *association rules* in the context of the relational model of data, and then introduce the definition of *association rule intention*.

4.1 Association Rules

The association patterns among attribute values are represented as association rules, where an association rule is an implication of the form:
lhs → *rhs,*
Each of the left had side (lhs) and right had side (rhs) is a set of attribute values, provided that
lhs ∩ *rhs* = □ .
For instance, {referral} → {long} is an association rule relating the attribute value MEMBERSHIP_DURATION.long to the attribute value HOW_INTRODUCED.referral. Each association rule has two metrics to measure its interestingness, *support* and *confidence*. The support of an association rule is the support of the itemset that contains all items in the rule, that is, the itemset containing the union of the items of the *lhs* and *rhs*. In other words,

Support (lhs → rhs) =
support (lhs ∪ rhs) = P (lhs ∪ rhs)

As an example, to find the support of the rule {referral} → {long}, we note that 5 out of 13 tuples in the relation of Table 2 contain both referral and long, therefore,

Support (referral → long) = *Support* {referral, long} = (5/13) X 100 = 38.5%

The confidence of the rule (*lhs → rhs*) is the percentage of tuples that contain *rhs* from those that contain *lhs*. In other words, *confidence* is the conditional probability *P(rhs | lhs)*. In a given relation, if there are 60 tuples that contain *lhs*, and out of those there are 40 tuples that also contain *rhs*, then, *confidence* (*lhs → rhs*) = (40/60) X 100 = 66.7%. *Confidence* can be expressed in terms of support as follows:

$$Confidence\ (lhs \rightarrow rhs) = \frac{support(lh\ s \cup rhs)}{support(lh\ s)} \times 100$$

4.2 Association Rule Intensions

An association rule intention is a rule template that is shared by multiple association rules. Similar to an itemset intention, an association rule intention is expressed in terms of attribute names (written in upper case letters) instead of actual data values. For example,

AGE → MEMBERSHIP_DURATION
is an association rule intention. The following association rules are possible instantiations of the above rule intension.

young → long	*young → short*
middle → long	*middle → short*
senior → long	*senior → short*

Generally, an association rule intention can be written as *LHS→RHS,* where each of *LHS* and RHS represents a set of attribute names, provided that *LHS* ∩ *RHS* = □. An association rule of the form *lhs → rhs* is said to be an *instantiation of (⊏)* an association rule intention of the form *LHS → RHS if lhs ⊏ LHS AND rhs ⊏ RHS* (the "instantiation of" or "⊏" for itemsets is formally defined in Section 3) . In this case we say that *(lhs→ rhs) ⊏ (LHS → RHS*). Formally this is defined as follows.

(lhs→ rhs) ⊏ (LHS → RHS) iff
(lhs ⊏ LHS) ∧ (rhs ⊏ RHS)

A user of a data mining tool can guide the association rule mining process by specifying the rule intensions of interest. The data mining process can then find association rules that are instantiations of the given rule intension, and that satisfy a minimum support and a minimum confidence. This simplifies the data mining process and improves performance, since the system should mine for only a subset of all possible association rules.

5. Associations among Attributes from Multiple Tables

Using our approach of applying itemset and association rule computations directly to the relational model as opposed to market basket data representation enables us to use many features that are readily available in relational database systems. One of those features is the ability to create database views.

When a user wants to mine for association rules between data of two attributes that exist in two different relations, it is possible to do that by simply creating a view that joins the two tables and then applying the data mining process to the view. To demonstrate with a concrete example, let the following be three relations in a medical clinic database.

DOCTOR (D_ID, D_NAME, YEARS_OF_EXP)
PATIENT (P_ID, P_NAME, ADDRESS)
VISIT (D_ID, P_ID, FEE, P_SATISFACTION)

In this database, VISIT is a relation that stores facts about every visit made by a patient to a doctor in the clinic. The attribute P_SATISFACTION records the patient's satisfaction with the visit and has two possible values 'satisfied' and 'unsatisfied.' YEARS_OF_EXP records the number of years of medical experience that the doctor has in the form of 5-year ranges, such as 1-5, 6-10, 11-15, etc.

Assume a user is interested in discovering the relationship between the years of experience that doctors have and the degree of satisfaction of their patients. In other words the user is interested in association rules whose intention is of the form:

YEARS_OF_EXP → P_SATISFACTION

The problem with such association rule is that the two attributes are located in two different relations. To alleviate the problem, a SQL statement can be written to create a view containing these two attributes. A relational view is considered as a virtual relation, and once created can be used just as a relation. Therefore, an association rule mining algorithm that can be applied to a relation can also be applied to a view. Below is the SQL statement syntax to create a view called EXPERIENCE_IMPACT, which contains the two desired attributes.

```
CREATE VIEW EXP_IMPACT as
SELECT YEARS_OF_EXP, P_SATISFACTION
FROM DOCTOR D, VISIT V
WHERE D.D_ID = V.D_ID
```

Mining can be performed on the view EXP_IMPACT to find all association rules that satisfy a minimum support and a minimum confidence and whose intention is of the form *YEARS_OF_EXP → P_SATISFACTION.*

6. Conclusion

In this paper, we introduced association rules in the context of the relational model of data as opposed to the market basket data representation. Market basket data representation is widely used in the research literature as a basis for association rule mining. However, in many cases, it is not straight forward to convert relational data to market basket data representation in order to perform association rule mining. In addition, the conversion process is a time consuming prerequisite especially for large quantities of data. We believe that association rules can be mined directly form data represented in the relational data model. In this paper, we showed how itemsets and association rules can be defined in the context of the relational model. We also introduced the concepts of *itemset intention* and *association rule intention*, and demonstrated how support and confidence of association rules can be computed in the relational context. Finally, we showed that if association rules mining is applied directly to relational data represented based on the relational model, the association rules that relate attributes from more than one relation can also be easily mined. This can be achieved by creating database views and mining these views.

Acknowledgment
The author would like to thank the Applied Science University in Amman, Jordan, for supporting this publication.

References

[1] Michael Hahsler and Kurt Hornik, "New probabilistic interest measures for association rules," Intelligent Data Analysis: an Internnationl Journal, 11(5):437-455, 2007.

[2] G. Liu, H. Lu, Y. Xu, J. X. Yu, "Ascending Frequency Ordered Prefix-tree: Efficient Mining of Frequent Patterns," Proc. 2003 Int. Conf. on Database Systems for Advanced Applications (DASFAA03), Kyoto, Japan, March 2003.

[3] Karam Gouda and Mohammed Zaki, "GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets," Data Mining and Knowledge Discovery: An International Journal, 11(3): 223-242, 2005.

[4] Y. Fu and J. Han, "Meta-Rule-Guided Mining of Association Rules in Relational Databases," Proc. First Int'l Workshop Integrating Knowledge Discovery with Deductive and Object-Oriented Databases (KDOOD '95), pages 39–46, Singapore, Dec. 1995.

[5] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arining, and T. Bollinger "The Quest Data Mining System." International conf. on Data Mining and Knowledge Discovery (KDD'96), pages 224-249, Portalnd, Oregon, 1996.

[6] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the International Conf. on Very Large Databases (VLDB'94), pages 487-499, Santiago, Chile, 1994.

[7] P. Berka, I. Bruha, "Discretization and Grouping: Preprocessing Steps for Data Mining," Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery, Pages: 239 – 245, 1998.

[8] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation" In *Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00)*, pages 1-12, Dallas, TX, May 2000.

[9] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers, March 2006.

[10] C. Date, *An Introduction to Database Systems, Eighth Edition*, Pearson Addison Wesley, 2003.

[11] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, *Fifth Edition,* Addison-Wesley, 2007.

Abdallah Alashqur obtained his Masters and Ph.D. degrees from the University of Florida. He worked for several years in the IT industry before he decided to come back to academia. Currently he is a faculty member in the Faculty of Information Technology at the Applied Science University in Amman, Jordan. His research interests include data mining and database systems.