IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol. 9, Issue.10, October 2021 E-ISSN: 2347-2693

Query Processing In Text Mining

N. BhanuPrakash[1], E. Kesavulu Reddy[2*]

[1,2]Dept. of Computer Science, S V University College of CM&CS, Tirupati-517502, Andhra Pradesh-India

Corresponding Author:ekreddysvu2008@gmail.com, Mobile: 9866430097.

DOI: https://doi.org/10.26438/ijcse/v9i10.1923 | Available online at: www.ijcseonline.org

Received: 18/Oct/2021, Accepted: 20/Oct/2021, Published: 31/Oct/2021

Abstract:-Companies often use relational database management systems (RDBMS) such as Oracle and Inform mix, to store their data persistently. The database technology developed and deployed in RDBMS is relatively mature. Besides efficient storage and retrieval, this technology provides many additional features such as concurrency control, recoverability, and high availability. Thirdly, the rigid structure of relational data makes it amenable to complex queries and analysis such as on-line analytical processing (OLAP), the predecessor of data mining. There are many different techniques and algorithms for relational data that can be classified as data mining. There are roughly four broad classes i.e. clustering, classification, sequence analysis, and associations. We consider data mining for structured data from a database perspective. As a consequence in association rules will be featured more prominently than the other three classes of mining problems. Query flocks are an elegant framework for a large class of data mining problems over relational data. The main features of query flocks are declarative formulation of a large class of mining queries. Systematic optimization and processing of such queries Integration with relational DBMS, taking full advantage of existing capabilities. This paper focus mainly on the declarative formulation of mining problems as query flocks.

Keywords:- RDBMS, Clustering, Query flocks, Query Optimization, Relational data, Classification, Clustering, sequence Analysis

I. INTRODUCTION

Data mining, from its inception, has targeted primarily relational data. There are numerous reasons that warrant this almost exclusive focus; here, we recount three of the most important ones. Systematic optimization and processing: The current state of the art in data mining of structured data is ad-hoc optimization techniques that only apply to specific problems and limited types of data. In group query, mining queries are optimized and processed systematically in the form of query of the existing capabilities of RDBMS. Group query, on the other hand, can be easily integrated with a relational database. Furthermore, the integration can be tightly coupled meaning that the query-processing capabilities of the database are utilized fully.

There are many different techniques and algorithms for relational data that can be classified as data mining. The underlying assumption behind clustering and classification is the a-priori existence of a model of which the actual data is just an observed instance. Association rules, on the other hand, are data-centric, and patterns that emerge do not have to be combined to derive a complete model. Furthermore, the amount of data that has been subjected to association-rule mining is several orders of magnitude larger than the amount of data normally used in classification and clustering. In a nutshell, association rules are data mining from a database perspective while classification, clustering, and sequence analysis have a machine-learning bias.

II. RELATIVE WORK

Association-rule mining is widely regarded as the association of data mining. Since its introduction in [1], the problem of mining association rules from large databases has been investigated in numerous studies. The topics range from improving the basic a-priori algorithm [2], to mining generalized or multilevel rules [3], to parallel algorithms [4] and incremental maintenance [5]. The vast majority of these studies share the basic a-priori technique based on level wise pruning. Recently, however, there has been some work on finding different kinds of association rules where the basic a-priori technique cannot be applied on [6], and [7]. The notion of a query with a filter condition representing a data mining problem has been proposed in the first but this proposal has very limited query form and a complex filter language involving set variables. However, the proposed query form is nothing more that basic association rules which limits the type of mining problems that can be expressed.
[8]. another proposal for a query with a filter condition is presented in [9]. We will use as our query language \conjunctive queries" [10], augmented with arithmetic, negation, and union. The filter condition will be expressed in a SQL-like manner over the result of the query. We will use Data log [11] notation to express conjunctive queries.

Consider a typical supermarket where every day thousands of shoppers come to the checkout registers with baskets of supermarket items. An observant store manager may note that many customers tend to purchase several specific items together, e.g., bread and milk, beer, chips, and salsa, or vodka and caviar. Furthermore, the store manager may notice that not only many people buy beer, chips, and salsa together, but also people rarely buy just beer and chips. In other words, customers who buy beer and chips are especially likely to buy salsa. We call such a pattern an association rule. The goal of association-rule mining is to discover such patterns automatically from large amounts of data.

Items: I = {beer; bread; chips; milk; salsa}
Baskets: B = {{bread; milk},{ bread; chips; ilk},{beer; chips},{beer; bread; chips; salsa}, {beer; chip; salsa},{beer; bread; milk}}

Figure 1.1: Simple example of market basket data

Definition: Let I = $\{i_1,i_2,..i_k\}$ be a set of k elements, called items. Let B = $\{b_{1,}\ b_2 \dots b_n\}$ be a set of n subsets of I. We call bi $\subseteq$ I a basket of items.

An association rule is intended to capture the extent of co-occurrence of two sets of items in the given basket data. The itemset P is associated with the itemset Q, and write P $\rightarrow$ Q, where P $\subseteq$ I and Q $\subseteq$ I. There are several quantities that measure the importance of an association rule. In the original definition [1], we have the following two:

$$\text{Support } (P \rightarrow Q) = \frac{\left|\{\mathrm{b_i} \left| (P \cup Q) \subseteq b_i \}\right|\right|}{n}$$

$$\text{Confidence } (P \rightarrow Q) = \frac{\left|\{\mathrm{b_i} \left| (P \cup Q) \subseteq b_i \}\right|\right|}{\left|\{b_i \left| P \subseteq b_i \}\right|\right|}$$

The support is the fraction of all n baskets that contain all items from both P and Q. The confidence is the fractions of the baskets which contain P that also contain Q. Note that both the confidence and support of an association rule are real numbers in [0; 1].

III. METHODOLOGY

A. A-priori Optimization Algorithm

There is an important optimization technique, called a-priori that makes the search for item sets with high support very efficient. The a-priori technique, introduced in [AIS93], is one of the main reasons for the apparent success and popularity of association rule mining. The key idea of a-priori is to use level wise pruning to reduce the number of item sets with potentially high support. From the definition of support, it follows immediately, that if an itemset P has high support than any subset of P also has high support.

```
Algorithm:  Input: I - set of k items
B - set of baskets
minSup - support threshold
Output: Fi - sets of frequent item sets of size i = 1…k
$C_1$ = I
i = 1
$F_j = \phi$ ; for j = 1…k
While $C_i \neq \phi$ do
Compute support (P) from B for all P ϵ $C_i$
$F_i$ = {P | P ϵ $C_i$ , support(P) >= minSup}
$C_{i+1}$ = generate _candidates (Fi)
i = i + 1
end while
return Fj for j = 1…k
```

Items, such as {bread}→{milk} there are $\theta(k^2)$ potential rules, where k is the number of items. There is an important optimization technique, called a-priori that makes the search for item sets with high support very efficient. The a-priori technique, introduced in [1], is one of the main reasons for the apparent success and popularity of association rule mining. The key idea of a-priori is to use level wise pruning to reduce the number of item sets with potentially high support. From the definition that an itemset P has high support than any subset of P also has high support.

B. Definition of Query Groups

Consider the given example of supermarket data. Suppose, we are interested in finding all frequent item sets of size 2, i.e., pairs of items. In principle, we can enumerate all possible pairs and for each pair {X;,Y } ask the query \How many baskets contain both X and Y ". Then, we can check whether the answer of each query is greater than the given support threshold. If so, we add the pair{X,Y} to our final result. We designate X and Y as parameters, then we have many identical queries except for the values of their parameters. Hence, the idea of a group of queries, or a group query. Thus, a query is the parameterized query that represents all possible simple queries, with instantiated parameters, and the filter condition that we apply to the answer of each simple query.

Formally, we define a query as

1. One or more predicates that represent the given data stored as relations.
2. A set of parameters.
3. A parameterized query.
4. A filter that specifies conditions that the result of each instantiated queries must satisfy in order for a given assignment of values to the parameters to be acceptable.

It is important to distinguish between parameterized queries and group query. A group query is a query about its parameters. The result of the flock is not the result of the parameterized query that is used to specify the group.

C. Query Groups

The representation of a query filter condition problem in the data mining. The key idea is to express both the query and the filter as logic statements. Thus, the filter can be as complex as the query. For example, the filter may state that one of the items in a market basket must be bread. In query flocks, the role of the filter is limited to a condition about the result of the query.

All are proposed that queries are very limited form but a complex filter language involving set of variables.
However, the proposed query form is nothing more that basic association rules which limits the type of mining problems that can be expressed.

We will use as our query language \conjunctive queries augmented with arithmetic, negation, and union. The filter condition will be expressed in a SQL-like manner over the result of the query. We will use Data log notation to express conjunctive queries. Data log has two major advantages specific to the query flock framework:

1. The notion of \safe query" for Data log is directly applicable to query optimizations for query flocks.
2. The generalization of the a-priori technique for query flocks and more complex optimization tricks are most apparent and intuitive when expressed in Data log.

In order to specify the query part of a query flock, in Data log terminology ([Ull88]), we need to provide the following:

1. Extensional predicates that represent the given data stored as relations.
2. A set of parameters, which we will always
   Denote with names beginning with R.
3. Intentional predicates expressed as conjunctive queries with added arithmetic and negation over the extensional predicates.

For the filter language we use SQL conditions similar to the ones in the HAVING clause [12]. A condition is an equality or inequality of two expressions. Each expression can involve the following:

1. Aggregate functions: COUNT, SUM, AVG, MIN, MAX.
2. Basic arithmetic: (+;; _; =).
3. Standard mathematical functions such as log; sqrt; abs; etc.
4. Constants (real numbers).
5. Attributes (columns) of intentional or extensional predicates.

IV. RESULTS AND DISCUSSION

MARKET BASKET ANALYSIS AS A QUERY GROUPS

We will consider the simplest market-basket problem as a query flock. We are given relation baskets (BID, Item) as the only extensional predicate representing the underlying data. Table 1.1 gives example contents of the baskets relation. Recall that market basket analysis is about finding those pairs of items R1 and R2 that appear in at least c baskets.

1.1. Table: Baskets Relation

Bid	Item
100	Bread
100	Milk
101	Bread
101	Chips
101	Milk
102	Fruit
102	Chips
103	Fruit
103	Bread
103	Chips
103	Salsa
104	Fruit
104	Chips
104	Salsa
105	Fruit
105	Bread
105	Milk

A. Query Groups.1. (Basic Market Baskets)

QUERY: Answer(B):- baskets(B,R1) AND
baskets(B,R2)
FILTER: COUNT (answer.B) >= 20

Example 1. Query Flock .1 finds pairs of items that appear in at least 20 baskets. For any values of R1 and R2, the query asks for the set of baskets B in which items R1 and R2 both appear. The answer relation for this pair of items is the set of such baskets. Then, the R1, R2

Table.1.2. Result of The Market-Basket Group Query

R1	R2
Fruit	Diapers
Bread	Milk
Milk	Bread
……..	…….

Filter condition requires that the set of such baskets number at least 20. The result of the query flock is thus the set of pairs of items (R1,R2) such that there are at least 20 baskets containing both items R1 and R2. Table 1 gives an example of the group query result.

V. MULTIPLE INTENTIONAL PREDICATES

The most natural query flocks, and indeed the flocks for which we have the most promising optimization techniques, involve support as the filter condition; Query Flock 1 is such a flock. It is possible to represent confidence, interest, and other conditions as Filters, using our SQL-like Filter language. However, it is necessary to allow the query portion of a flock to produce several relations as its result. Thus, we need multiple intentional predicates so that we can express the filter condition. Furthermore, we can have several different filter conditions, e.g. high support and high confidence.

```
Query Flock 2. (Market Baskets)
QUERY: Answer1 (B) :- baskets(B,$1) AND
          baskets (B,$2)answer2(B) :- baskets(B,$1)
FILTER:
2 * COUNT(answer2.B) >= COUNT(answer1.B)
 (high confidence) COUNT(answer.B) >= 20 (high
  support)
```

A. Expressions with Query Groups

One of the main objectives of query groups is to allow the declarative formulation of a large class of mining queries. We discuss the other typical mining problems such as classification, clustering, and sequence analysis phrased as query flocks.

- Classification

A typical problem in classification is to find the best k attributes in order to predict accurately the class of certain instances. Here, we consider the following modified problems.
Suppose we have the following data:

attributes(InstanceID, AttributeName, AttributeValue)

class(InstanceID, ClassName)

We want to find all pair of attributes and their corresponding values such that knowing the two values, we can predict the class of an instance, with 80% accuracy (based on the underlying data). The following group expresses this problem:
Query Flock 3.(Classification)

```
QUERY:
Answer1 (I) :-
attributes (I,R1,R2) AND
attributes(I,R3,R4) AND
class(I,R5)
Answer2 (I) :-
attributes(I,R1,R2) AND attributes(I,R3,R4)
FILTER:COUNT(answer1.I)>=0.8* OUNT(answer2.I)
```

The result of Query Flock 3 is a set of quintuples (R1,R2,R3,R4,R5). The interpretation of this result is that if we know for a particular instance that the value of attributes R1 and R2 are R3 and R4 respectively, then we can guess the class of the instance to be R5 with 80% accuracy.

- Clustering

Consider the following simple clustering problem. We are given a set of two dimensional points and we want to divide them into four regions, using one horizontal and one vertical line, such that each of the four regions contains at least 1/5 of all points. The points are given as the following relation: points(x, y). The given below query group expresses the problem of choosing a horizontal line Y=R2 and Vertical line at X=R1

```
QUERY:
answer1(P,Q) :- points(P,Q) AND P<=R1 AND Q<R2
answer2(P,Q) :- points(P,Q) AND P>R1 AND Q<=R2
answer3(P,Q) :- points(P,Q) AND P>=R1 AND Q>R2
answer4(P,Q) :- points(P,Q) AND P<R1 AND Q>=R2

FILTER:
COUNT(answer1(*) >= COUNT(points(*)/5
COUNT(answer2(*) >= COUNT(points(*)/5
COUNT(answer3(*) >= COUNT(points(*)/5
COUNT(answer4(*) >= COUNT(points(*)/5
```

- Sequence Analysis

One of the basic problems in sequence analysis is to identify a subsequence that occurs frequently in a given sequence of events. We model this problem with the following example. A events has some information relation of some events occurring sequentially.

events(Sequence Number, Event Type)

The problem is to find a frequent subsequence of event types R1, R2, R3 such that R2 occurs within two events after R1, and R3 occurs within two events after R2. Query group expresses this problem, again calculate frequent to mean at least 20 occurrences.

```
Query Group 4. (Frequent Subsequence)
QUERY: answer(L) :- events(L, $1) AND events(M,
                    $2) AND events(N, $3) AND
                   L >= M-2 AND M >= N-2
                   L < M AND M < N
FILTER: COUNT (answer.L) >= 20
```

VI. CONCLUSION and FUTURE SCOPE

We introduced the query flock framework for mining relational data. Query groups allow a declarative formulation of large class of data mining queries. Basket analysis to apply to any query flock in our Class. By using the concept of query safety, we described the possible sub queries that could be used to exploit the a-priori idea, and we then suggested several techniques for further limiting the search for query plans. These techniques are either static heuristics, where we enumerate a class of plans and estimate the cost of each, based on available size estimates for relations, or dynamic, and the size of Intermediate results before deciding whether or not to apply a filtering step.

Monotone Filter Condition techniques are not discussed in this paper. By monotone we mean that if the condition is true for a given set then it must also be true for any superset of the original set. Examples include certain COUNT, MIN, MAX, SUM (in the case of non-negative numbers) conditions. As a simple example, we can extend the traditional market basket problem, whose flock appeared in Fig. 2 to a *weighted market basket*, where the baskets B have weights, a associated through a relation importance(B,W).

REFERENCES

[1]. R. Agrawal, T. Imilienski, and A. Swami. *"Mining association rules between sets of items in large databases"* in the Proceedings of ACM SIGMOD International Conference on Management of Data, pages 207{216, **May 1993.**

[2]. S. Brin, R. Motwani, D. Tsur, and J. Ullman. "*Dynamic itemset counting and implication rules for market basket data."* in the Proceedings of ACM SIGMOD International Conference on Management of Data, pages 255{264, Tucson,Arizona, June **1997.**

[3]. R. Srikant and R. Agrawal. "*Fast algorithms for mining association rules"* in the Proceedings of the 21th International Conference on Very Large Data Bases, pages 407{419, Zurich, Switzerland, September **1995.**

[4]. E. Han, G. Karypis, and V. Kumar *"Scalable parallel data mining for association rules"* in the Proceedings of ACM SIGMOD International Conference on Management of Data, pages 277{288, Tucson, Arizona, June **1997.**

[5]. D. W. Cheung, J. Han, V. Ng, and C. Y. Wong. "*Maintenance of discovered association rules in large databases: An incremental updating technique*" in the Proceedings of ICDE, pages 106{114, New Orleans, Louisiana, February **1996.**

[6]. R. Motwani, E. Cohen, M. Datar, S. Fujiware, A. Gionis, P. Indyk, J. Ullman, and C. Yang, " *Finding interesting associations without support pruning"*In Proceedings of ICDE, San Diego, California, March **2000.**

[7]. S. Fujiware, R. Motwani, and J. Ullman " *Dynamic miss-counting algorithms: Finding Implication and similarity rules with con dence pruning", i*n the Proceed-ings of ICDE, San Diego, California, March **2000.**

[8]. H. Mannila *"Methods and problems in data mining"* in the Proceedings of International Conference on Database Theory, pages 41{55, Delphi, Greece, January **1997.**

[9]. R. Ng, L. Lakshmanan, J. Han, and A. Pang. "*Exploratory mining and pruning optimizations of constrained associations rules*" in the Proceedings of ACM SIG-MOD International Conference on Management of Data, pages 13{24, Seattle,Washington, June **1998.**

[10]. A. Chandra and P. Merlin "*Optimal implementation of conjunctive queries in relational databases*" in the Proceedings of 9th Annual ACM Symposium on the Theory of Computing, pages 77{90, Boulder, Colorado, May **1977.**

[11]. J.D. Ullman "*Principles of Database and Knowledge-Base Systems" in* Volume I - Fundamental Concepts. Computer Science Press, Rockville, Maryland, **1988.**

[12]. J.D. Ullman and J. Widom. A First Course in Database Systems. Addison Wesley, Reading, Massachusetts, **1997.**

[13]. B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi keyword fuzzy search over encrypted data in the cloud," in IEEE INFOCOM, **2014.**

[14]. V.Kaltsa, K. Avgerinakis, A. Briassouli, I. Kompatsiaris and M. Strintzis, "Dynamic texture recognition and localization in machine vision for outdoor environments," Computers in Industry, **vol. 98, 2018.**