International Journal of Research in Advent Technology, Vol.2, No.5, May 2014
E-ISSN: 2321-9637

Mining Frequent Item and Item Sets Using Fuzzy Approach

Ms. Poonam A. Manjare[1], Mrs R.R.Shelke[2]
Computer Science and Engineering Department[1, 2]
Email: manjarepoonam@gmail .com[1] ,rajeshrishelke@rediffmail.com[2]

Abstract-Data mining is the central step in a process called knowledge discovery in databases, namely the step in which modeling techniques are applied. Several research areas like statistics, artificial intelligence, machine learning, and soft computing have contributed to its arsenal of methods. Data mining is an increasingly important technology for extracting useful knowledge hidden in large collections of data. The proposed work presents the design of mining frequent items from dataset. In proposed work, however, we focus on fuzzy methods information mining, and dependency analysis. Fuzzy approaches can play an important role in data mining, because they provide comprehensible. In addition, the approaches studied in data mining have mainly been oriented at highly structured and precise data. However, we expect that the analysis of more complex heterogeneous information source like texts, images, rule bases etc. will become more important in the near future. Therefore we give an outlook on information mining, which we see as an extension of data mining to treat complex heterogeneous information sources, and argue that fuzzy systems are useful in meeting the challenges of information mining.

Index Terms -Data Mining, Frequent Item, Frequent Item sets, Fuzzy slices

1. INTRODUCTION

Information retrieval and data mining are two components of a same problem, the search of information and knowledge extraction from large amounts of data, very large databases or data warehouses. In information retrieval, the user knows approximately what he looks for, for instance an answer to a question, or documents corresponding to a given requirement in a database. The search is performed in text, multimedia documents (images, videos, sound) or in web pages. Tranmedia information retrieval takes advantage of the existence of several media to focus on a more specific piece of information, for instance using sound and speech to help retrieving sequences in a video. The main difficulty lies in the identification of relevant information, i.e. the closest or the most similar to the user's need or expectation. The concept of relevance is very difficult to deal with, mainly because it is strongly dependent on the context of the search and the purpose of the action launched on the basis of such expected relevant information. Asking the user to elicit what he looks for is not an easy task and, the more flexible the query-answer process, the more efficient the retrieval. This is a first reason to use fuzzy sets in knowledge representation to enable the user to express his expectations in a language not far from natural. The second reason lies in the approximate matching between the user's query and existing elements in the database, on the basis of similarities and degrees of satisfiability. In data mining, the user looks for new knowledge, such as relations between variables or general rules for instance.

Mining data streams is a very important research topic and has recently attracted a lot of attention, because in many cases data is generated by external sources so rapidly that it may become impossible to store it and analyze it offline. Moreover, in some cases streams of data must be analyzed in real time to provide information about trends, outlier values or regularities that must be signaled as soon as possible. The need for online computation is a notable challenge with respect to classical data mining algorithms [1], [2]. Important application fields for stream mining are as diverse as financial applications, network monitoring, security problems, telecommunication networks, Web applications, sensor networks, analysis of atmospheric data, etc. A further difficulty occurs when streams are distributed, and mining models must be derived not only for the data of a single stream, but for the integration of multiple and heterogeneous data streams [3]. Two important and recurrent problems regarding the analysis of data streams are the computation of frequent items and frequent item sets from transactional datasets. The first problem is very popular both for its simplicity and

because it is often used as a subroutine for more complex problems. The goal is to find, in a sequence of items, those whose frequency exceeds a specified threshold. When the items are generated in the form of transactions, *sets* of distinct items, it is also useful to discover frequent sets of items. A k-item set, i.e., a set of *k* distinct items, is said to be frequent if those items concurrently appear in a specified fraction of transactions.

2. RELATED WORK

The analysis of data streams has recently attracted a lot of attention owing to the wide range of applications for which it can be extremely useful. Important challenges arise from the necessity of performing most computation with a single pass on stream data, because of limitations in time and memory space. Stream mining algorithms deal with problems as diverse as clustering and classification of data streams, change detection, stream cube analysis, indexing, forecasting, etc [**8**]. In the propose work, a major need is to identify frequent patterns in data streams, either single frequent elements or frequent sets of items in transactional databases. A rich survey of algorithms for discovering frequent items is provided by Cormode and Hadjieleftheriou [**4**]. In proposed work, the discussion focuses on the two main classes of algorithms for finding frequent items. Counter-based algorithms have their foundation on some techniques proposed in the early 80s to solve the Majority problem [**9**], i.e., the problem of finding a majority element in a stream, using a single counter. Variants of this algorithm were devised, sometimes decades later, to discover items whose frequencies exceed any given threshold. LossyCounting is perhaps the most popular algorithm of this type [**5**]. The second class of algorithms computes a *sketch*, i.e., a linear projection of the input, and provides an approximated estimation of item frequencies using limited computing and memory resources. Popular algorithms of this kind are CountSketch [**10**] and CountMin [**7**], and the latter is adopted in this paper. Advantages and limitations of sketch algorithms are discussed in [13]. Important advantages are the notable space efficiency (required space is logarithmic in the number of distinct items), the possibility of naturally dealing with negative updates and item deletions, and the linear property, which allows sketches of multiple streams to be computed by overlapping the sketches of single streams. The main limitation is the underlying assumption that the domain size of the data stream is large, but this is true in many significant domains. Even if modern single-pass algorithms are extremely sophisticated and powerful, multi-pass algorithms are still necessary either when the stream rate is too rapid, or when the problem is inherently related to the execution of multiple passes, which is the case, for example, of the frequent item sets problem. Single-pass algorithms can be forced to check the frequency of 2- or 3-itemsets, but this approach cannot be generalized easily, as the number of candidate *k*-item sets is combinatorial, and it can become very large when increasing the value of *k* [**4**]. Therefore, a very promising avenue could be to devise hybrid approaches, which try to combine the best of single- and multiple-pass algorithms. A strategy of this kind, discussed in [**6**], is adopted in the mining architecture presented in this paper. The analysis of streams is even more challenging when data is produced by different sources spread in a distributed environment. A thorough discussion of the approaches currently used to mine multiple data streams can be found in [11]. The paper distinguishes between the *centralized* model, under which streams are directed to a central location before they are mined, and the *distributed* model, in which distributed computing nodes perform part of the computation close to the data, and send to a central site only the models, not the data. Of course, the distributed approach has notable advantages in terms of degree of parallelism and scalability. An interesting approach for the continuous tracking of complex queries over collections of distributed streams is presented in [3]. To reduce the communication overhead, the adopted strategy combines two technical solutions: (i) remote sites only communicate to the coordinator concise summary information on local streams (in the form of sketches); (ii) even such communications are avoided when the behavior of local streams remains reasonably stable, or predictable: updates of sketches are only transmitted when a certain amount of change is observed locally. The success of this strategy depends on the level of approximation on the results that is tolerated. A similar approach is adopted in [12]: here stream data is sent to the central processor after being filtered at remote data sources. The filters adapt to changing conditions to minimize stream rates while guaranteeing that the central processor still receives the updates necessary to provide answers of adequate precision.

3. FUZZY BASED MINING OF FREQUENT ITEMS

Objectives of proposed work are summarized as follows:

3.1. Fetching Records Which Matches Our Search Criteria

Mining data is very important because data generated by external sources must be analyzed in real time for the computation of item and item sets.

Steps to fetch records

- In this all records would fetch from database.
- After that get the search from user.
- Make the list of all records from database.
- At each record concatenation of name and its description is done and split that record by space, after that matching will be done with search query

3.2. Data Analysis

This would used to analyze the fetch data and find the data which the user had searched previously.

3.3. Fuzzy Slices based Searching for Liked Items

Here data would be distributed in fuzzy sets. A fuzzy set provides a natural basis for the theory of possibility.

Steps include in fuzzy slices based searching

- The items from database which the user likes may have been purchased by the user, so we get the all matching items which the user has purchased.
- After getting the list of purchased items, match these items with input search query and add it to the output fuzzy set.
- Now the searched result from the step 2 are checked in the database for other linked items, these items are present in liked list.
- All the items which are in the liked list are viewed by the user, these items are also included in view list.

3.4. Frequent View List

This would include the list of items which are viewed by other user frequenly.

3.5. Recommendation

Here data would be recommending through fuzzy sets.

3.6. Result Evaluation and system operation

Result evaluation would be done through fetched data from database, analysis of fetched data which determine frequent item and item sets and distributed data from fuzzy sets.

4. CONCLUSION

Stream mining system is a contribution in the field and it aims at solving the problem of computing frequent items and frequent item sets from distributed data streams by exploiting a hybrid single pass/multiple-pass strategy. We assumed that stream sources, though belonging to different domains, are homogenous, so that it is useful to extract knowledge from their union. Beyond presenting the system architecture, we described a prototype that implements it and discussed a set of experiments performed in a real Grid environment. The experimental results confirm that the approach is scalable and can manage large data production by using an appropriate number of miners in the distributed architecture.

REFERENCES

[1] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Mining data streams: A review,” ACM SIGMOD Record, vol. Vol. 34, no. 1, 2005.

[2] C. C. Aggarwal, Data Streams: models and algorithms. Springer, 2007.

[3] G. Cormode and M. Garofalakis, “Approximate continuous querying over distributed streams,” ACM Transactions on Database Systems, vol. Vol. 33, no. 2, 2008.

[4]R. Jin and G. Agrawal, “An algorithm for in-core frequent itemset mining on streaming data,” in 5th IEEE International Conference on Data Mining ICDM, Houston, Texas, USA, 2005, pp. 210–217.Conference on Data Mining ICDM, Houston, Texas, USA, 2005, pp. 210–217.

[5]A. Wright, “Data streaming 2.0,” Communications of ACM (CACM), vol. Vol. 53, no. 4, 2010.

[6]G. Manku and R. Motwani, “Approximate frequency counts over data streams,” in International Conference on Very Large Data Bases, 2002.

[7] G. Cormode and S. Muthukrishnan, “An improved data stream summary: The count-min sketch and its applications,” J. Algorithms, vol. Vol. 55, 2005.

[8]C. Aggarwal, “An introduction to data streams,” in Data Streams: Models and Algorithms, C. Aggarwal, Ed. Springer, 2007, pp. 1–8.

[9] M. Fischer and S. Salzburg, “Finding a majority among n votes: solution to problem 81-5,” J. Algorithms, vol. 3, no. 4, pp. 376–379, 1982.

[10] M. Charikar, K. Chen, and M. Farach-Colton, “Finding frequent items in data streams,” in Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP), 2002.

[11] A. G. Srinivasan Parthasarathy and M. E. Otey, “A survey of distributed mining of data streams,” in Data Streams: Models and Algorithms, C. Aggarwal, Ed. Springer, 2007, pp. 289– 307.

[12] C. Olston, J. Jiang, and J. Widom, “Adaptive filters for continuous queries over distributed data streams,” in SIGMOD ’03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, San Diego, California, 2003.

[13] C. C. Aggarwal and P. S. Yu, "A survey of synopsis construction in data streams," in Data Streams: Models and Algorithms, C. Aggarwal, Ed. Springer, 2007, pp. 169–207.