International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 5 39 – 42

Frequent pattern Mining Implementation on Social network for Business Intelligence

Ujwala Mhashakhetri
Department of Computer Science & Engineering
Rajiv Gandhi College of Engineering, Research & Technology
Chandrapur, India
mujwala1990@gmail.com

Dr. Rahila sheikh
Department of Computer Science & Engineering
Rajiv Gandhi College of Engineering, Research & Technology
Chandrapur, India
Rahila.patel@gmail.com

***Abstract*—**In the world of online business the social media is found to be the biggest player in field of marketing, advertisement and many other. Customers in the online market are going very choosy in the selection of product, they check for the better product & for that most of the time they go through reviews of product. Sometimes a feature might be interesting for one, while it does not make that impression for someone else. So, it is necessary to identifying the target product with particular features & it is a tough task which is difficult to achieve with existing functionality. In this paper, we present a frequent pattern mining algorithm to mine a customer reviews and extract product features. Our experimental results indicate that the algorithm outperforms the old pattern mining techniques used by previous researchers.

Keywords-*Frequent pattern mining ; Feature extraction; Business intelligencesolution.*

*****

I. INTRODUCTION

Social Network analysis and mining has great importance in today's world because of the huge amount of data available on the social network. Many researches has been done on social network mining to solve different problems but there is a gap between technique developed by research community & their deployment in the real world applications. Therefore the potential impact of such application is still unexplored [1]. A business process classification framework can be used to map the techniques used to solve problems in social network mining to business processes. The tools and techniques developed for analyzing and mining social networks can be used in a wide range of processes across the enterprise.

The APQC Process Classification Framework. There is a large body of research on business process management, and several business process classifications. The PCF was developed as an open standard for improvement through process management and benchmarking, without considering industry, size, or geography. The PCF [1] organizes operating and management processes into 12 enterprise-level categories, including process groups and over 1,000 processes and associated activities. In all the categories social network analysis and mining has a crucial role [2]. The 3rd category from 12 categories " Marketing and selling" include activity called social CRM which is a new emerging area used for trend spotting, detecting future business opportunities as well as reputation monitoring . Social networks like Twitter, Facebook are used by organizations, consultancies and companies to interact with customers. During this their aim is to make more followers/customers for their solutions and/or products. Customer comments help users to shape and market their products. But it is not easy to read each and every comment of customer, to know which attribute or component of product is getting which type of feedbacks & how many numbers of feedbacks. Therefore, many researchers have proposed various techniques to discover such information automatically. Opinion Mining or Sentimental Analysis determines whether the comment is of positive, negative or neutral orientation [14]. Product feature extraction is very difficult for sentiment analysis, because the opinion orientation identification is significantly affected by the target features [15]. Therefore, in this paper we focus on product feature extraction from customer reviews. Specifically, we present new frequent pattern mining algorithm to apply for frequent feature extraction from the review sentences. Existing frequent pattern mining algorithms which are used for discovering Frequent Item sets from the transaction datasets are Apriori algorithm[3], Apriori TID algorithm[3], FP-Growth algorithm[4], Eclat algorithm[5], dEclat algorithm[6], Relim algorithm[7], H-mine algorithm[8], LCM Freq algorithm[9], Pre Post and Pre Post+ algorithms[10], FIN algorithm[11]. Algorithms for performing targeted and dynamic queries about association rules and frequent item sets are Item set-Tree [12], Memory-Efficient Item set-Tree[13].

The key contribution of this paper is to develop a novel frequent pattern mining approach based on Item-Set Tree Data structure as it is found to be more effective for incremental data. The rest of the paper is organized as fallows. In section 2 introduces the problem definition & basic concepts, section 3 describes the algorithm & system framework, section 4 contains Experimental results & Evaluation, section 5 conclude the paper & the related references.

II. PROBLEM DEFINATION

This section first defines the general problem of feature identification of reviews and then it gives the problem which we aim to solve. Following are some basic definitions.

A. Itemset Tree:

An itemset-tree is a special data structure that can be used for performing efficient queries about itemsets and association rules in a transaction database without having to generate all of them beforehand.

B. Transaction

A transaction is simply a set of distinct items

C. Product feature

A Product feature refer to all the components, qualities or physical characteristics of a product for example size, color, weight, speed, etc.

D. Opinion sentence

An opinion sentence is a sentence that contains minimum one product feature and its corresponding opinion word.

E. Explicit and implicit feature

An explicit feature is a feature of a product which is directly talked about in review sentence. An implicit feature is a feature of a product that is not explicitly mentioned in the sentence and it can be implied.

F. Frequent and infrequent feature

A feature *f* is said to be frequent if it appears in maximum number of the review sentences. *f* is called infrequent if it is only appeared in a minimum number of reviews.

Here, the problem is how to identify feature in review both explicit and implicit. Another problem is for one dataset of products with negative or neutral word can be positive for other product. A common unsupervised approach that has proposed by many researchers is based on association mining technique. They focus on the nouns or noun phrases which is supposed to be frequently occurs in the review dataset are most likely to beconsidered as product features. In [16], Hu *et al.* used an NL Procesor to parse all their views and produce the part-of-speech tag for each word. After identifying nouns they ran an association miner which is based on Apriori algorithm to find frequent item sets that are likely to be frequent features. This method is simple and efficient and gives reasonable results. However, this technique has some major shortcomings. Apriori algorithm tests combination of the items, without considering of the items ordering. For instance, the words “dvd” and ”player” may be occurred in 14 transactions (sentences) as ”player dvd” while 87 transactions contain “dvd player”. The algorithm cannot recognize the difference between the two situations and it returns only one possible combination such as “palyer dvd” with totally 101 occurrences. However, depending on the chosen threshold, the item “player dvd” may be considered as an infrequent item and it is not expected to be listed here. Moreover, in case that there exist a large number of frequent patterns, Apriori have to take many scans of large databases and generate huge number of candidates which reduces the performance of the system. Our work focuses on handling the above problems with the previous work by applying a more efficient frequent pattern mining algorithm with the Item set Tree Data structure.

TABLE I. EXAMPLE HOW TO USE ITEMSET TREE DATASTRUCTURE

Transaction ID’s	Items
T1	{1,4}
T2	{2,5}
T3	{1,2,3,4,5}
T4	{1,2,4}
T5	{2,5}
T6	{2,4}

For example, we could insert the following 6 transactions (t1,t2...t6) into an itemset-tree.

The result of the insertion of these six transactions is the following itemset-tree.

{} sup=6
[2] sup=3
[2 5] sup=2
[2 4] sup=1
[1] sup=3
[1 2] sup=2
[1 2 4] sup=1
[1 2 3 4 5] sup=1
[1 4] sup=1

The root is the empty itemset {} and the leafs are {2, 5}, {2, 4}, {1 2 4},{1 2 3 4 5} and {1, 4}.

Once an itemset-tree has been created, it is possible to update it by inserting a new transaction. For example, in this example provided in the source code, we update the previous tree by adding a new transaction {4, 5}. The result is this tree:

{} sup=7
[2] sup=3
[2 5] sup=2
[2 4] sup=1
[1] sup=3
[1 2] sup=2
[1 2 4] sup=1
[1 2 3 4 5] sup=1
[1 4] sup=1
[4 5] sup=1

Next, it is shown how to query the tree to determine the support of a target itemset efficiently. For example, if we execute the query of finding the support of the itemset {2}, the support is determined to be 5 because 2 appear in 5 transactions.

After that the source code offers an example of how to use the itemset tree to get all itemsets that subsume an itemset and to get their support. For example, if we use the itemset {1 2} for this query the result is:

[1 2] supp:2
[1 2 3] supp:1
[1 2 4] supp:2
[1 2 5] supp:1
[1 2 3 4] supp:1
[1 2 3 5] supp:1
[1 2 4 5] supp:1
[1 2 3 4 5] supp:1

An itemset-tree has the nice property of being incremental, which means that new transactions can be added to an existing itemset tree very efficiently without having to rebuild the tree from scratch. An itemset-tree also has the property of being compact.An itemset-tree is built by inserting a set of transactions into the tree.

III. PROPOSED SYSTEM

Proposed The system architecture of feature extraction system is given in Figure 1 and each system component is detailed subsequently. The input to the system is a product review/ comment dataset including a large number of reviews on products. It is a free dataset which is available for download at *http://www.cs.uic.edu/~liub/FBS/CustomerReviewDat.zip*. The output of the system will be obtained after passing the following five phases.

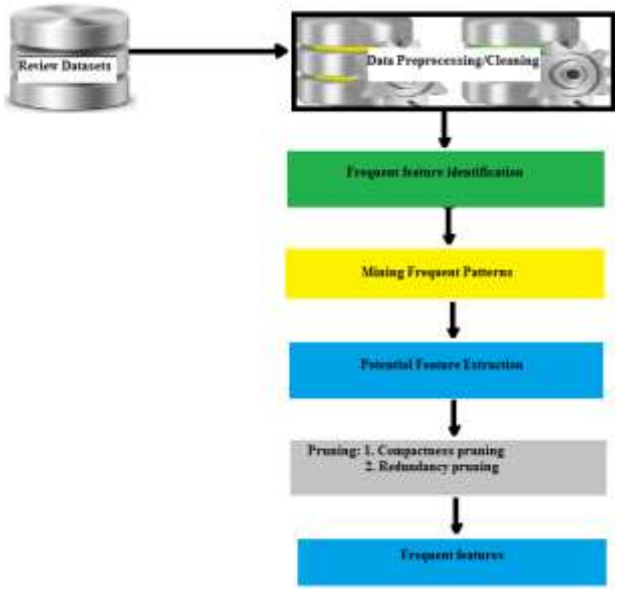

Figure 1. System framework .

Figure 2.

A. Preprocessing

It includes Data cleansing the process of detecting and correcting (or removing) corrupt or inaccurate which are incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and then replacing, modifying, or deleting this dirty data or coarse data.

The actual process of data cleansing may involve removing typographical errors or validating and correcting values against a known list of entities. It also includes stop word removal and stemming Data cleansing may also involve activities like, harmonization of data, and standardization of data. For example, harmonization of short codes (St, rd etc.) to actual words (street, road). Standardization of data is a means of changing a reference data set to a new standard, ex, use of standard codes.

B. Part-of-Speech-Tagging

As the only focused part of the sentences in our work is nouns or noun phrases, we apply a Part-Of-Speech tagger that we developed in PHP to identify the role of the words within the sentences. The following shows a tagged sentence after removing its stop words:

Original sentence: "The camera is very easy to carry."
Tagged sentence: camera/NN easy/JJ carry/VB

Each sentence is filtered by the identified noun tags and the result is saved in our review dataset.

C. Frequent Feature Identification

All the documents include the reviews about same product. Moreover, a product feature is a noun or noun phrase which is appeared in review sentences. Given the fact, it can be inspired that the nouns with high frequency can most likely be considered as feature words. Frequent pattern mining techniques tend to determine multiple occurrence of the same item. So we have taken the advantage of such techniques in our work in order to find frequent nouns or noun phrases as the potential feature words. In frequent pattern mining if the frequent item have a high relative support such as 10% or more then the projected database is dense so the itemset tree is not bushy so it is beneficial for use otherwise Unlike Hu's work, we can apply a faster and space-preserving frequent pattern mining algorithm called H-Mine(Mem) [17] to work with large sparse datasets

D. Pruning

Association mining algorithms does not consider the position of the items in a given transaction. Thus, after running the algorithm on a sequence of words as an input transaction, it generates a number of candidates that may not be genuine features. On the other hand, in a natural language the words that are appeared together in a specific order usually deliver a particular meaning and they are most likely considered as meaningful phrases. Referring to the above discussion we define a compact feature are a feature phrase that its words do not appear together in the sentence. In this paper we remove non-compact features in the following manner:

For each sentence in the review database
{
If (a feature phrase found)
{
For each feature in the sentence
{
Measure the distance between every two words;
If (words distance > 3)
Remove the feature from the list;
}
}
}

Suppose that an identified feature is life battery. The algorithm goes through the database and checks if there exist, at least one occurrence of the two words life and battery which appear in a sentence with distance of 3. If it cannot find a sentence, the feature will be removed from the list. Focusing on features that contain only one word, we also apply another technique to remove redundant features. As a definition, the number of sentences that feature *ftr* is appeared in and there are no superset of *ftr* is called *pure support* of *ftr*. Given the definition, a redundant feature refers to a feature which is subset of another feature phrase and has a pure support lower than minimum p-support. In this work we set the minimum value of p-support into 3 and calculate p-support of every feature. Then those features which have p-support lower than minimum are ignored.

IV. Experimental Results and Evaluation

A To evaluate the efficiency of our feature extraction & product review system we compare our frequent pattern mining algorithm results with Apriori that has been used by [16]. We first report our experimental results on the performance of H-mine & algorithm for Itemset tree in comparison with Apriori with the dense and sparse datasets and then evaluate the accuracy of the system while using these two algorithms.

A. Performance measure

Performance measure is the execution time of the algorithms. All tests were done on a desktop with an Intel® Core™2 Duo *processor, 4GB Ram,* and a fresh installed *Windows 7* Professional. Figure 2 illustrates average execution by running Apriori and H-Mine on our sparse review dataset.

Figure 3 illustrate average execution by running Apriori and Itemset Tree algo on our dense review dataset.

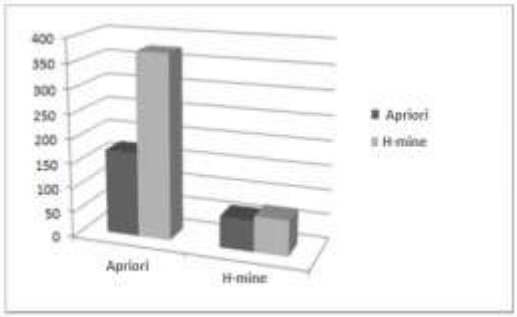

Figure 3. System framework .

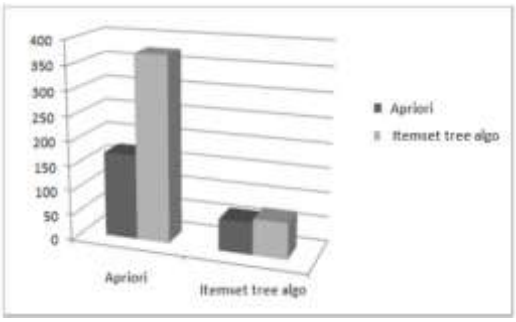

Figure 4. System framework .

From the figure, we can see that H-mine is much faster than Apriori because of the traversing strategy with the sparse dataset & Itemset Tree algorithm is much faster than Apriori with the dense dataset. It tries to divide the search space and mine the partitions locally while Apriori follows test-and-generate strategy to mine dataset. Moreover, H-Mine scans database only one time to find the frequent itemsets. Then a tree view of the data is constructed in the main memory and the algorithm starts to explore the tree in a depth-first search manner. In second case Itemset Tree algorithms with dense dataset have high relative support which leads to database compression due to common prefix path. A comparison on the two results reveals that there is an inverse relation between the execution time of the algorithms in sparse and dense datasets.

V. CONCLUSION

In this paper, we used a pattern mining algorithm called Item set tree algorithm & H-mine to discover features of products from reviews. It is able to deal with two major problems: 1) taking many scans of large databases to generate frequent item sets, and 2) lack of recognizing transposition of the words while generating new item sets. In this work we only focused on those features that frequently appear in the review sentences. Latter on these frequent features can be used as criteria on which positive, negative & neutral comments/reviews are calculated for product review. Our experimental results indicate that our method outperforms the old pattern mining technique.

REFERENCES

[1] Francesco Bonchi, Carlos Castillo, Aristides Gionis & AlenjandroJaims, “Social Network Analysis & Mining For Business Applications”, In ACM Transactions on Intelligent Systems and Technology, Vol. 2, No. 3, Article 22, Publication date: April 2011.

[2] www.apqc.org/OSBCdatabase

[3] Rakesh Agrawal , Ramakrishnan Srikant, “Fast Algorithms for Mining Association Rules”, In J.B. Bocca, M. Jarke, and C. Zaniolo, editors, Proceedings 20th International conference on Very Large Data Bases, pages 487-499. Morgan Kaufmann, 1994.

[4] JIAWEI HAN, JIAN PEI, YIWEN YIN, RUNYING MAO, “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach”, Data Mining and Knowledge Discovery, 8, 53–87, 2004.

[5] Mohammed J. Zaki, “Scalable Algorithms for Association Mining”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 12, NO. 3, MAY/JUNE 2000.

[6] Mohammed J. Zaki and Karam Gouda, “Fast Vertical Mining Using Diffsets”, presented at 9th International Conference on Knowledge Discovery & Data Mining, Washington, DC, 2003.

[7] Christian Borgelt, “Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination”, published in Proceeding of 1st international workshop on open source data mining: Frequent pattern mining implementations, pages 66-70, 2005.

[8] JIAN PEI, JIAWEI HAN, HONGJUN LU, SHOJIRO NISHIO, SHIWEI TANG and DONGQING YANG, “H-Mine: Fast and space-preserving frequent pattern mining in large databases”, IIE Transactions (2007) 39, 593–605.

[9] [9] Takeaki Uno , Masashi Kiyomi , Hiroki Arimura, “LCM ver. 3: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets”, FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004.

[10] DENG ZhiHong, WANG ZhongHui & JIANG JiaJian, “A new algorithm for fast mining frequent itemsets using N-lists”, Sci China Inf Sci, 2012.

[11] Zhi-Hong Deng , Sheng-Long Lv, “Fast mining frequent itemsets using Nodesets”, Applied Soft Computing 07/2015

[12] Miroslav Kubat, Alaaeldin Hafez, Vijay V. Raghavan, Jayakrishna R. Lekkala, “ Itemset Tree for Targeted Association Querying”, published in IEEE Transactions on Knowledge and Data Engineering Volume 15 Issue 6, Page 1522-1534, November 2003.

[13] Philippe Fournier-Viger , Eserance Mwamikazi , Ted Gueniche and Usef Faghihi, “MEIT: Memory Efficient Itemset Tree for Targeted Association Rule Mining”, published in the 9th International Conference on Advanced Data Mining and Applications (ADMA 2013), At Hangzhou, China.

[14] B. Liu, "OPINION MINING," In: Encyclopedia of Database Systems, 2004.

[15] R. Hemalatha, A. Krishnan and R. Hemamathi, "Mining Frequent Item Sets MoreEfficiently Using ITL Mining," in *3rd International CALIBER*, Ahmedabad, 2005.

[16] K. Dave, S. Lawrence and D. M. Pennock, "Mining the peanut gallery: opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th international conference on World Wide*, New York, 2003.