Rare Pattern Mining

Ian Smith

April 15, 2017

University of Manitoba

# Introduction

Data mining is defined as the non-trivial extraction of implicit, previously unknown, and potentially useful information from data . Rare Pattern mining is the extraction of patterns that occur very infrequently within a database. Currently, much of the research available on data mining focuses on frequent pattern mining; however, rare pattern mining is an exceedingly interesting, growing field within computer science . My research will demonstrate that a frequent pattern mining algorithm, the Frequent Pattern Growth algorithm, can be modified to retrieve interesting rare itemsets that can then be studied. We hope to also show that our Rare Pattern Growth algorithm can do this efficiently.

There are quite a few different applications for rare pattern mining. For example, rare pattern mining can be employed for fraud detection. In a database of credit card transactions, the rare patterns could turn out to be fraudulent, and the credit card company could use a rare pattern mining algorithm to detect and prevent these transactions from harming their customers. A company that does this might be able to attract more customers by promising them protection.

There are two important ways to determine whether or not a pattern can be considered rare: its support, and its confidence level. Support represents the number of time the items that make up the set occur together in the database . Confidence represents a proportional value that shows how frequently one part of the itemset occurs with the other(s) . Because an itemset with very little support is likely, if not guaranteed to have a low confidence measure, it is only necessary to consider the support value of the itemset to determine whether or not it is rare.

The Frequent Pattern Growth algorithm uses a divide-and-conquer approach toward mining frequent patterns. The frequent single items of each transaction are added one at a time into a tree data structure, and then each frequent itemset is mined from the tree using a recursive function. My Rare Pattern Growth will operate in much the same way. The more frequent rare items will be added into the tree first, and mined from it last.

The organization of this report is as follows: first I will examine existing literature in data mining that focuses on rare itemset mining, as well as Frequent Pattern Growth. Second, I will describe my algorithm to mine rare itemset using diagrams and pseudo-code. I will also be examining the efficiency of using VIPER. Third, I will analyze the results of my algorithm. And finally I will conclude with an examination of what I learned from running the algorithm, and how it could be applied to the data mining field as a whole.

# Literature Review

Aggrawal and Han wrote one an excellent, comprehensive book about the study of frequent pattern mining. This resource has proved invaluable in examining Frequent Pattern Growth for this project, and other areas of data mining, as it successfully provides an overview of the different frequent pattern mining methods .

Weng proposed an Apriori-based mining approach called Fuzzy Apriori Rare Itemset Mining [FARIM], for mining “specific rare itemsets consisting of quantitative data” .[[1]](#footnote-1) Weng proposed using FARIM for low test or quiz scores in a school setting; if there was a student, or a group of students struggling with class content, then determining exactly what it is they are struggling with would go a long way in finding a solution [5]. Weng believed that his approach would be more successful if it included clustering and classification methods, and if the support parameter was inferred from the data.

Hemalatha, Vaidehi, and Lakshmi wrote about finding rare itemsets in data streams, as opposed to static datasets . To that end, they proposed an algorithm for finding Minimal Infrequent Patterns from Data Streams, defined three measures for outlier detection, and created a Minimal Infrequent Pattern based Outlier Detection algorithm. They found, among other things, that their methods were well suited for extracting useful data from sensor data streams and identifying meaningful outliers from those streams.

Wu, Chen, and Chang wrote about Attribute-Oriented Induction (AOI), and proposed using AOI to mine negative generalized knowledge from datasets . Their reasoning has to do with medical data; for example, if only a few Taiwanese people were infected with the H1N1 flu virus the number of people that are Taiwanese and have contracted H1N1 will be very small, and not considered a frequent itemset. However, if few Taiwanese contracted H1N1, then that might indicate that the Taiwanese were somehow resistant to the disease.

Agrawal and Agrawal presented an overview of how data mining techniques could be used to detect anomalies in datasets . Their Classification approach was to build a model based on the normal behavior of the system, and then feed testing data into that model in order to determine which datasets were anomalous . They tried various different Clustering approaches including k-Means, k-Medoids, and other approaches. They found that hybrid approaches, which combine Clustering and Classification based anomaly detection systems, had the best chance at finding anomalous behaviors in Intrusion Detection Systems.

Lin, Liao, and Chen actually wrote about using the Frequent Pattern Growth algorithm to find frequent itemsets, and in particular to reduce the number of candidate itemsets examined by the algorithm, and reducing the number of times it is necessary to scan the entire database . This is because the Apriori algorithm requires scanning the database repeatedly, and Dynamic Hashing and Pruning algorithm improves the performance of Apriori and lowers the cost of database scanning. But FP-Growth goes even further to improve the performance of Frequent Pattern Mining.

Lin, Lao, and Chen then propose using an Improved FP-Growth algorithm to improve the performance of FP-growth . They do this in part by using an address-table structure to lower the complexity of mapping frequent 1-itemsets in an FP-tree, and by using a hybrid FP-tree mining method that reduced the need to rebuild conditional FP-trees. Their simulation shows that their algorithm improved the performance of FP-growth by an order of magnitude in terms of execution time.

Cagliero and Garza actually use FP-Growth in a similar to fashion to what I am using it for, rare pattern mining .

Bhattacharyys, Jha, Tharakunnel, and Westland wrote about how data mining could be utilitzed to detect or prevent credit card fraud . According to them $4 billion was lost in 2008. They went on to identify two different approaches for detecting fraud: supervised and unsupervised. Supervised fraud detection methods detect fraudulent transactions by estimating based on samples of both fraudulent and legitimate transactions in order to classify new transactions as one or the other . In unsupervised fraud detection models outliers are identified as potential cases of fraud and sought out.

Bhattacharyya et al then went on to examine two different mining techniques for finding fraudulent transactions: random forests and support vector machines, together with logical regression . They concluded that the random forests demonstrated better performance and greater efficiency compared to support vector machines . They suggested that future study should focus on the different varieties of fraudulent behavior, for example the difference between stolen and counterfeit credit cards.

Yu, Sheikholeslami, and Zhang wrote about finding outliers in very large datasets . This is because, as they put it, “Modern companies are awash in data on customers, clients, suppliers, and industry trends” . Their contribution, called FindOut, was intended to detect outliers in complicated data patterns of various densities. It did so using signal-processing techniques and a novel deviation outlier detection approach. It was also successful in indentifying various percentages of outliers in large datasets .

Adda, Wu, White, and Feng wrote about finding rare patterns . They modeled their approach on the Apriori algorithm, and used it to detect abnormal usage in web applications. They found that their approach was flexible and able to detect suspicious behaviors not seen before .

# Description

The software prototype operates in a number of steps. First, it reads in the transaction database, then it creates an array of Transaction objects, then it removes the non-rare items from those transactions, then it adds those transactions one-at-a-time to a tree structure, then it analyzes that tree structure to find all the rare patterns in the database.

## Transaction Stage

The transaction stage is fairly simple to understand. The software reads in the transaction database. The database is stored in the form of a simple text file with each line indicating a transaction, and the first l-line indicating the number of transactions in the file. Each one of these lines is used to initialize a Transaction object, and each transaction contains one or more items. These items may or may not be rare in the database at this point.

## Pruning Stage

The Pruning Stage is where the non-rare items are removed from all the transactions in the database. The database itself is not altered, but any number of Transaction objects that the software uses could be. This stage is completed in several steps. First, every item in the database is added to an Itemset object, along with its support. Then the items with too much support in the database are removed from the itemset, and then those items are removed from the Transactions.

In the first stage, it was necessary to two more classes: the Item class, and the Itemset class, which stores an array of Items, and associated functionality. The Item class is used to store the name of a database item, and its support. The software loops through every transaction, and if an item is found that is not yet in the Itemset, then it is added in with a support of one, if it was already present, then its support is incremented by one.

Next, the actual pruning begins .The software removes each item that has too much support in the database from the Itemlist. The maximum support is determined in the software itself, but it is always some fraction of the total number of transactions in the database. Any item that is not rare in the database cannot be part of a rare itemset, according to a modification of the Apriori property for frequent patterns. This well known data mining property states that every subset of a frequent itemset must also be frequent.

Once there is a definitive list of rare single items in the database, it is only a matter of looking at each Transaction, and removing the non-rare items from the Transaction. This usually requires replacing the Transaction’s item array with a new one, then deleting the old one.

Lastly, the items in the Itemset, and then the Transactions are sorted so that the items which occur most frequently in the database will be added to the tree first. This might seem a little counter-intuitive for a rare pattern tree, but it is necessary for the analysis stage where the rare patterns are collected from the tree.

## Construction Stage

In this stage, the transactions are used to construct a Rare Pattern tree.

# Analysis

# Conclusion

# Bibliography

|  |  |
| --- | --- |
| [1] | D. Yu, G. Sheikholeslami and A. Zhang, "Finding Outliers in Very Large Datasets," *Knowledge and Information Systems,* vol. 4, pp. 387-412, 2002. |
| [2] | L. Cagliero and P. Garza, "Infrequent Weighted Itemset Mining using Frequent Pattern Growth," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 4, pp. 903-915, April 2014. |
| [3] | M. Adda, L. Wu, S. White and Y. Feng, "Pattern Detection with Rare Itemset Mining," *International Journal on Soft Computing, Artificial Intelligence and Applications,* vol. 1, no. 1, pp. 1-17, August 2012. |
| [4] | Y.-Y. Wu, Y.-L. Chem and R.-I. Chang, "Mining negative generalized knowledge from relational databases," *Knowledge-Based Systems,* vol. 24, no. 1, pp. 134-145, February 2011. |
| [5] | C.-H. Weng, "Mining fuzzy specific rare itemsets for education data," *Knowledge-Based Systems,* vol. 24, no. 5, pp. 697-708, July 2011. |
| [6] | C. S. Hemalatha, V. Vaidehi and R. Lakshmi, "Minimal infrequent pattern based approach for mining outliers in data streams," *Expert Systems with Applications,* vol. 42, no. 4, pp. 1998-2012, March 2015. |
| [7] | S. Agrawal and J. Agrawal, "Survey on Anomaly Detection using Data Mining Techniques," *Procedia Computer Science,* vol. 60, pp. 708-713, 2015. |
| [8] | K.-C. Lin, I.-E. Liao and Z.-S. Chen, "An improved frequent pattern growth method for mininig association rules," *Expert Systems with Applications,* vol. 38, no. 5, pp. 5154-5161, May 2011. |
| [9] | S. Bhattacharyya, S. Jha, K. Tharakunnel and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Elsevier,* vol. 50, no. 3, pp. 602-613, February 2011. |
| [10] | C. C. Aggarwal and J. Han, Eds., Frequent Pattern Mining, Springer Cham Heidelberg, 2014. |
| [11] | R. S. Debreceny and G. L. Gray, "Data mining journal entries for fraud detection: An exploratory study," *International Journal of Accounting Information Systems,* vol. 11, no. 3, pp. 157-181, September 2010. |

1. The Apriori algorithm is a frequent pattern mining algorithm where frequent single-item-sets are combined to create larger frequent itemsets, and then the database is scanned to determine the support of the new itemsets. This process continues until there are no more itemsets that can be combined [10] [↑](#footnote-ref-1)