

# Backward Support Computation Method for Positive and Negative Frequent Itemset Mining

Mrinmoy Biswas Akash, Indrani Mandal, Md. Selim Al Mamun

Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Mymensingh, Bangladesh  
Email: mamun0013@gmail.com

**How to cite this paper:** Akash, M.B., Mandal, I. and Al Mamun, Md.S. (2023) Backward Support Computation Method for Positive and Negative Frequent Itemset Mining. *Journal of Data Analysis and Information Processing*, 11, 37-48.  
<https://doi.org/10.4236/jdaip.2023.111003>

**Received:** November 7, 2022

**Accepted:** February 3, 2023

**Published:** February 6, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Association rules mining is a major data mining field that leads to discovery of associations and correlations among items in today's big data environment. The conventional association rule mining focuses mainly on positive itemsets generated from frequently occurring itemsets (PFIS). However, there has been a significant study focused on infrequent itemsets with utilization of negative association rules to mine interesting frequent itemsets (NFIS) from transactions. In this work, we propose an efficient backward calculating negative frequent itemset algorithm namely *EBC-NFIS* for computing backward supports that can extract both positive and negative frequent itemsets synchronously from dataset. *EBC-NFIS* algorithm is based on popular *e-NFIS* algorithm that computes supports of negative itemsets from the supports of positive itemsets. The proposed algorithm makes use of previously computed supports from memory to minimize the computation time. In addition, association rules, *i.e.* positive and negative association rules (PNARs) are generated from discovered frequent itemsets using *EBC-NFIS* algorithm. The efficiency of the proposed algorithm is verified by several experiments and comparing results with *e-NFIS* algorithm. The experimental results confirm that the proposed algorithm successfully discovers NFIS and PNARs and runs significantly faster than conventional *e-NFIS* algorithm.

## Keywords

Data Mining, Positive Frequent Itemset, Negative Frequent Itemset, Association Rule, Backward Support

## 1. Introduction

Association rules mining have been studied successfully and extensively in many application domains such as market analysis and telecommunications [1] [2] [3] [4]. As digitalized data are increasing due to human dependency on technologies,

the world has an abundance of data which if processed properly can turn into one of the most valuable resources of information and knowledge. Specific methods and algorithms are required to extract knowledge from data, which is simply known as data mining. Along with classification and clustering, frequent pattern mining is one of the major sub-fields of data mining and knowledge discovery. Purpose of frequent pattern mining is to obtain itemsets from data based on common associations. It can be useful to find patterns that frequently appear in data or to discover items often existing together. Applications of frequent pattern mining range from supermarket sales analysis to bioinformatics.

Algorithms like Apriori [5], FP-Growth [6] can mine what is referred to in the literature as positive association rules (PAR). To mine positive associations from a data set the items with strong positive correlations are considered. However, itemsets with strong negative correlation called negative association rule (NAR) may also be as informative as positive ones. For example, from supermarket sales data NAR mining may reveal products scarcely bought together. In behavior analysis lack of specific behaviors may help understand human psychology.

Formally a positive association rule is of the form  $X \Rightarrow Y$  where  $X$  and  $Y$  are separate entities. The rule  $X \Rightarrow Y$  can be crudely translated into if  $X$  exists in an itemset and  $Y$  also exists in it. For NARs, either one or both of the related entities are negated. i)  $\neg X \Rightarrow Y$ , ii)  $X \Rightarrow \neg Y$  or iii)  $\neg X \Rightarrow \neg Y$  are the possible negative associations, whereas rules like  $\neg X \Rightarrow Y$  says if  $X$  does not exist in an itemset and  $Y$  exists in it. An example can be  $\text{Pepsi} \Rightarrow \neg \text{Coke}$ , because people who buy one brand of cold-drinks from a store are less likely to buy another along with it.

Negative relations among items were first mentioned in [7], but the first significant work on it was [8] which combined previously discovered positive associations with domain knowledge to reduce the search space. Reducing the search space enabled it to discover fewer but more interesting negative associations. There have been few other notable works in the domain. In [9] authors defined four types of association rules from Piatetsky-Shapiro's argument and used statistical dependency and conditional probability to discover association rules. In 2006, an algorithm named PNAR based on the Apriori method was proposed [10]. The algorithm exploits the upward closure property of negative association rules to find all valid positive-negative associations in the support-confidence framework.

The same year, authors in [11] presented an approach for mining positive and negative association rules which used chi-squared test and multiple confidences. Combining the Multiple Level Minimum Support (MLMS) model with a new measure called Valid Association Rules based on Correlation-coefficient and Confidence (VARCC) which combines correlation coefficient and minimum confidence a new algorithm (PNAR\_MLMS) was proposed in [12]. Authors in [13] proposed an algorithm using correlation coefficient with support and confidence to mine both PFIS and NFIS. The coefficients need continuous updating while processing.

Because both positive and negative association rules are relevant, many algorithms focus on finding both simultaneously. The task of discovering (negative or positive) association rules can be divided into two parts. The first one is to mine frequent itemsets like A, B, C or A, B,  $\neg C$ , the latter is to generate association rules from them. The process of generating association rules mostly depends on the discovery of frequent itemsets and generally follows the same method everywhere. Frequent itemset extraction is what usually defines the difference in performance between two algorithms. However, the hidden nature of negative itemsets makes the mining task very complex. Authors often define their own definitions and parameters in their work. Besides, negative frequent itemsets (NFIS) are numerous in number. Even for a considerably small data set the search space becomes very large. Most NAR mining algorithms are therefore time-consuming.

This work proposes EBC-NFIS based on *e-NFIS* algorithm [14] that can discover NFIS from pre-calculated positive frequent itemsets (PFIS). It is clear that each NFIS is nothing but a recombination of a PFIS where one or more items are negated. *e-NFIS* first generates all possible negative itemsets from the permutation of positive itemsets. Then, it uses the inclusion-exclusion principles to calculate itemset support. This work attempts to speed up the second step of *e-NFIS* with an aim to reduce overall computation time. EBC-NFIS processes negative candidate itemsets in a systematic order to calculate supports of current step based on supports calculated in previous steps. This reduces large computation time of *e-NFIS* algorithm and determines itemset support in constant runtime instead of exponential. The experimental results demonstrate that EBC-NFIS performs significantly faster than *e-NFIS*.

The rest of the paper is organized as follows. Section 2 presents problem definition and theories related to this work. Section 3 and Section 4 present the proposed EBC-NFIS algorithm and experimental results respectively. Finally Section 5 concludes this paper with some future works.

## 2. Preliminaries

In this section, we first define the problem of backward support computation for EBC-NFIS. Then, we present generation of PFIS and NCIS along with working principle of conventional *e-NFIS* algorithm.

### 2.1. Problem Definition

Let  $I = \{I_1, I_2, I_3, \dots, I_n\}$  be a set of  $n$  distinct terms called positive items,  $NI = \{\neg I_1, \neg I_2, \neg I_3, \dots, \neg I_n\}$  be a set of  $I$ 's corresponding non-occurring items called negative items. Each positive item has its own negative counterpart. Similarly, each negative item has its positive counterpart. Here,  $I_1$  is called the positive partner of  $\neg I_1$ , and  $\neg I_1$  is called the negative partner of  $I_1$ .

Let  $D = \{t_1, t_2, t_3, \dots, t_n\}$  be a transactional database where each transaction  $t$  contains items from the set of positive items  $I$  such that  $t$  is a subset of  $I$  i.e.  $t \subset I$ . Each transaction is associated with a unique identifier, called  $t_{ID}$ . A non-

empty subset of  $\{I \cup NI\}$  is called an itemset. The number of items in an itemset  $X$  is the length of the itemset, denoted by  $length(X)$ . An itemset with length  $k$  is presented as  $k$ -itemset. We define  $count(X)$  as the number of transactions containing itemset  $X$  in transactional database,  $D$ . An association rule for itemsets  $X$  and  $Y$  is defined as  $X \Rightarrow Y$  where  $X, Y \subset \{I \cup NI\}$  and  $X \cap Y = \emptyset$ .  $X$  and  $Y$  are called *antecedent* and *consequent* of the association rule respectively.

Association rules have some standard measures that present quality and significance of the association. In this work, we employed two popular measures namely *support* and *confidence*.

- **Support:** The most popular framework for eliminating non-interesting itemsets is support calculation. The support of an itemset can be thought of as the probability of it being in a randomly chosen transaction. Mathematically the support of an item  $I_j$  can be written as,

$$support(I_j) = \frac{count(I_j)}{|D|} \quad (1)$$

Here  $|D|$  is the size of the transactional database  $D$ . In the support framework an itemset is considered to be *interesting* if its support is above a given threshold value called *minimum support* which is expressed as  $supp_{min}$  in this work. While mining frequent itemsets, an itemset  $I_j$  will be eliminated if  $support(I_j) \leq supp_{min}$ .

- **Confidence:** The confidence of an association rule is an indication of how often the rule has been found to be true. Confidence represents the percentage of transactions in database  $D$  containing  $X$  that also contain  $Y$  and denoted as follows:

$$conf(X \Rightarrow Y) = \frac{P(XY)}{P(X)} \quad (2)$$

The confidence value indicates how reliable the rule is. The higher the confidence value, the more likely the rules occur in a database if it is known that all rules are contained in that database. An association rule is considered to be interesting when its confidence value is greater than user defined minimum confidence. The minimum confidence is represented as  $conf_{min}$  throughout this paper.

Association rules having the support value greater than user defined minimum support,  $supp_{min}$ , and confidence greater than user defined minimum confidence,  $conf_{min}$  are called valid association rules. Therefore, by a valid association rule, we mean any expression of the form  $X \Rightarrow Y$  where,  $X \in \{A, \neg A\}$ ,  $Y \in \{B, \neg B\}$ ,  $A, B \subset I$ ,  $A \cap B = \emptyset$  and

- 1)  $support(X \Rightarrow Y) \geq supp_{min}$
- 2)  $con(X \Rightarrow Y) \geq conf_{min}$

## 2.2. Generation of Positive Frequent Itemsets (PFIS)

PFIS can be derived using an algorithm like Apriori [5] or FP-Growth [6]. Apri-

ori is a level based algorithm which first generates candidate itemsets and prunes non-interesting ones. FP-growth replaces the data set with a tree like data structure called FP-Growth tree and skips the candidate generation part entirely. Unlike Apriori, FP-Growth algorithm does not scan the whole data set multiple times and therefore performs significantly better for large data sets. Other notable positive frequent pattern mining algorithms include rapid association rule mining (RARM) [15] and ECLAT [16]. Like FP-Growth, RARM uses a novel tree data structure and can accelerate the mining process even for low minimum support. ECLAT uses a vertical database format. In this work, we employed Apriori algorithm to derive FPIS from transactional database.

### 2.3. Generation of Negative Candidate Itemset (NCIS)

NCIS can be generated from FPIS by permutation and negation of items. For example, given a positive itemset  $\{I_a, I_b, I_c\}$  the following negative itemsets can be produced:

- $\neg I_a I_b I_c$
- $I_a \neg I_b I_c$
- $I_a I_b \neg I_c$
- $\neg I_a \neg I_b I_c$
- $\neg I_a I_b \neg I_c$
- $I_a \neg I_b \neg I_c$
- $\neg I_a \neg I_b \neg I_c$

The conventional e-NFIS algorithm ignores itemsets where each member is negative. This work however considers even them as interesting enough to discover.

### 2.4. Support Calculation for NCIS

In this work, we employed support calculation method presented in [17] for negative itemsets. The basic idea is *inclusion-exclusion principle* of set theory.

Let,  $I$  be the set of all  $n$  items in a transactional database  $D$ . Suppose, we have a candidate negative itemset,  $NIC = \{x_1, x_2, x_3, \dots, x_p, \neg y_1, \neg y_2, \neg y_3, \dots, \neg y_q\}$  where  $x_i \in I, y_i \in I$  and  $1 \leq p, q \leq n$ . We can also write  $X = \{x_1, x_2, x_3, \dots, x_p\}$  and  $Y = \{y_1, y_2, y_3, \dots, y_q\}$ . Therefore,  $NIC = X \neg Y$  and its support can be calculated as below:

$$\begin{aligned}
 s &= support(NIC) \\
 &= support(X \neg Y) \\
 &= support(X \neg y_1 \neg y_2 \neg y_3 \dots \neg y_q) \\
 &= support(X) - \sum_{1 \leq i_1 \leq q} support(X y_{i_1}) + \sum_{1 \leq i_1, i_2 \leq q} support(X y_{i_1} y_{i_2}) + \dots \\
 &\quad + (-1)^{q-1} \sum_{1 \leq i_1, i_2, \dots, i_{q-1} \leq q} support(X y_{i_1} y_{i_2} \dots y_{i_{q-1}}) + (-1)^q support(XY)
 \end{aligned} \tag{3}$$

For example, the support of an itemset  $I_1 I_2 I_3$  can be calculated as,

$$\begin{aligned} support(I_1 I_2 I_3) = & support(I_1) - \{support(I_1 I_2) + support(I_1 I_3)\} \\ & + support(I_1 I_2 I_3) \end{aligned}$$

## 2.5. Working Principle of e-NFIS

The e-NFIS algorithm is widely used in negative itemset mining that generates NFIS from PFIS instead of scanning whole database [14]. The algorithm works in three steps:

- 1) Step1: Mine PFIS using traditional PFIS algorithms, such as Apriori or FP-Growth.
- 2) Step2: Generate NCIS from PFIS derived in Step1. Generation of NCIS from PFIS is described in 2.3.
- 3) Step3: Calculate support for NCIS. Supports of NCIS can be calculated using inclusion-exclusion principle as described in 2.4.

It should be noted that as NCIS is generated from PFIS, any subset of PFIS used in the support calculation must also be frequent and therefore will exist among the discovered itemsets. We employed e-NFIS algorithm with backward calculation of NCIS to derive NFIS in this work.

## 3. Proposed Backward Support Computation Algorithm

In this section, we describe proposed method of support computation for NCIS using backward calculations. Here, we define an itemset  $m$ -negative itemset if it contains exactly  $m$  number of negative items. For example,  $I_1 \neg I_2 \neg I_3$  is a 2-negative itemset. Computation of NCIS support increases with increasing value of  $m$  i.e. starting from 1-negative itemsets, then 2-negative itemsets and so on. Let, a candidate negative itemset  $NIC = \{x_1, x_2, x_3, \dots, x_p, \neg y_1, \neg y_2, \neg y_3, \dots, \neg y_q\}$ . An itemset  $W = \{x_1, x_2, x_3, \dots, x_p, \neg y_1, \neg y_2, \neg y_3, \dots, \neg y_{q-1}\}$  can be generated by excluding the last element  $\neg y_q$ . The support of newly generated  $NIC = W \neg y_q$  can be computed if the support of  $W$  is known as follows:

$$support(NIC) = support(W \neg y_q) = support(W) - support(W y_q) \quad (4)$$

For example, the support of an itemset  $I_1 \neg I_2 \neg I_3$  can be calculated as:

$$support(I_1 \neg I_2 \neg I_3) = support(I_1 \neg I_2) - support(I_1 \neg I_2 I_3) \quad (5)$$

All items from the power set of  $Y = \{y_1, y_2, y_3, \dots, y_q\}$  are required in computation. As  $count(Y) = q$  and  $|P(Y)| = 2^q$ , the time complexity of the support calculation step of e-NFIS is  $O(2^q)$  where  $q$  is the number of negative items in itemset  $NIC$ . Sacrificing space for time, the proposed EBC-NFIS can perform this step in constant or  $O(1)$  time because addition of only two values is required. Achieving from exponential to constant runtime is a huge improvement in performance. As the size of the data set increases naturally, the number of total items should also increase. Therefore, conventional e-NFIS runs significantly slow for those cases, where the performance of EBC-NFIS will not deteriorate. The proposed EBC-NFIS algorithm is summarized in **Algorithm 1**.

**Algorithm 1.** Proposed backward support calculation for EBC-NFIS.

---

**Input:**  $D$ : dataset,  $supp_{min}$ : minimum support,  $conf_{min}$ : minimum confidence  
**Output:**  $NFIS$ : negative frequent itemset,  $PNAR$ : positive and negative association rule  
**Method:**  
 $S \leftarrow$  take sample ( $D$ )  
 $PFIS \leftarrow$  Apriori( $D, supp_{min}$ )  
 $L_{max} \leftarrow \max(\text{length}(X) : X \in PFIS)$   
 $NFIS \leftarrow \emptyset$   
 $PNAR \leftarrow \emptyset$   
 $NCIS[m] \leftarrow \emptyset$   
 $m \leftarrow 1$   
**while**  $m \leq L_{max}$  **do**  
  **for each**  $X$  **in**  $PFIS$  **do**  
    **if**  $\text{length}(X) = 1$  **then**  
       $\text{support}(\neg X) \leftarrow |D| - \text{support}(X)$   
       $NCIS[m] \leftarrow NCIS[m] \cup (\neg X)$   
      **if**  $\text{support}(\neg X) \geq supp_{min}$  **then**  
         $NFIS \leftarrow NFIS \cup (\neg X)$   
        **if**  $\text{conf}(\neg X) \geq conf_{min}$  **then**  
           $PNAR \leftarrow PNAR \cup (\neg X)$   
        **end**  
      **end**  
    **end**  
    **else**  
      **for any itemset**  $NIC = x_1x_2x_3 \cdots x_p\neg y_1\neg y_2\neg y_3 \cdots \neg y_{(m-1)}\neg y_m$  **do**  
         $W \leftarrow x_1x_2x_3 \cdots x_p\neg y_1\neg y_2\neg y_3 \cdots \neg y_{(m-1)}$   
         $\text{support}(NIC) \leftarrow \text{support}(W) - \text{support}(Wy_m)$   
         $NCIS[m] \leftarrow NCIS[m] \cup (\neg NIC)$  ;  
        **if**  $\text{support}(NIC) \geq supp_{min}$  **then**  
           $NFIS \leftarrow NFIS \cup (\neg NIC)$   
          **if**  $\text{conf}(NIC) \geq conf_{min}$  **then**  
             $PNAR \leftarrow PNAR \cup (\neg NIC)$   
          **end**  
        **end**  
      **end**  
    **end**  
  **end**  
  Delete  $NCIS[m - 2]$  if exists to save memory ;  
   $m = m + 1$  // increase  $m$   
**end**  
**return**  $NFIS$  and  $PNAR$

---

## 4. Experimental Results

In this section, we present experimental results of EBC-NFIS algorithm for different datasets. First, we briefly describe hardware and software platforms and give a brief description of standard datasets used for the experiments. Then, we present experimental results and evaluate efficiency of proposed EBC-NFIS algorithm by comparing with e-NFIS algorithm [14].

### 4.1. System and Data Set Description

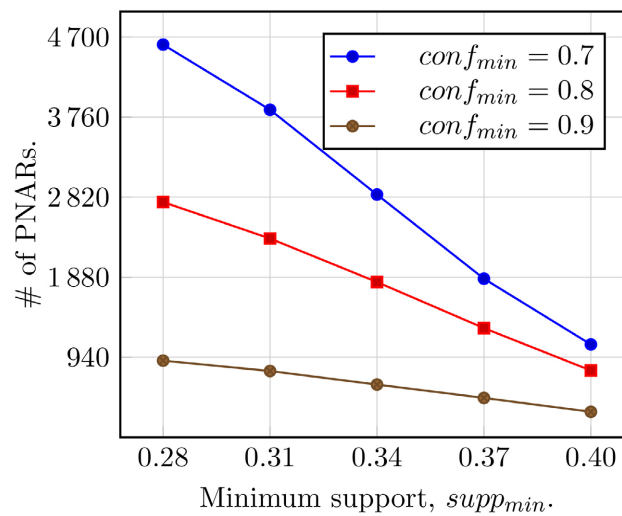
All the experiments have been performed on a system with software summarized in **Table 1**. In this work, we used three standard datasets namely *Breast Cancer* dataset, *Mushroom dataset* and *Congressional voting 84*. The datasets are collected from [18] that is an online repository of large data sets which encompasses a wide variety of data types, analysis tasks, and application areas. **Table 2** summarizes the number of transactions and number of items of both the datasets.

**Table 1.** Experiment environment.

Processor	Intel Core i5-4570 CPU @ 3.20 GHz
Memory	8 GB
Operating System	Ubuntu LTS 18.10
Software	Python3

**Table 2.** Datasets for experiments.

Dataset	# of transitions	# of items
Breast Cancer	286	9
Mushroom	2124	23
Congressional voting 84	435	16

**Figure 1.** Positive and negative association rules for different  $supp_{min}$  and  $conf_{min}$  for breast cancer dataset.

## 4.2. Result for Different Minimum Supports and Confidence

We conducted this experiment to find the number of positive and negative association rules (PNARs) for different minimum support,  $min_{sup}$  and confidence,  $conf$  using proposed EBC-NFIS algorithm. The experiment results based on the data sets of breast cancer, mushroom and congressional voting 84 are shown in **Figures 1-3** respectively. The experimental results indicate that the number of PNARs found by proposed algorithm increases for all the datasets when minimum support,  $supp_{min}$  decreases. Similarly, the number of PNARs increases when minimum confidence,  $min_{conf}$  decreases for all the datasets which is a fair characteristics of association rules searching algorithm.

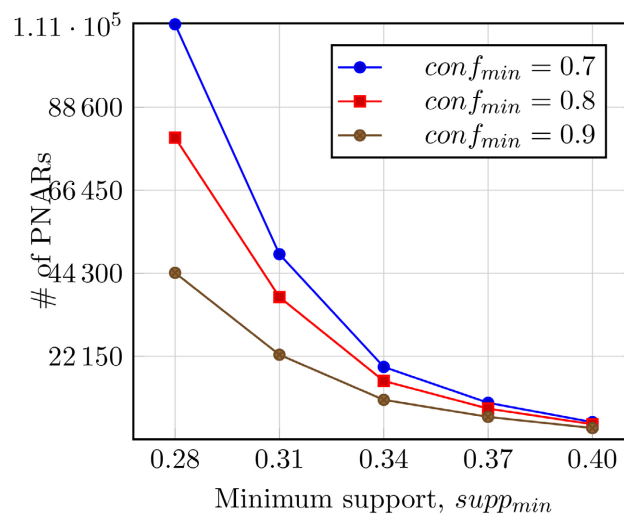
## 4.3. Comparison between e-NFIS and EBC-NFIS

In this section, we evaluated proposed EBC-NFIS algorithm by comparing running time with conventional e-NFIS algorithm. We run both e-NFIS and EBC-NFIS

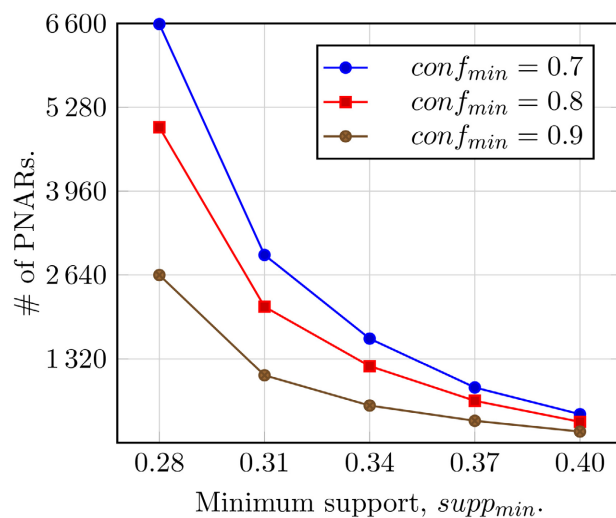


on both breast cancer and mushroom data set to observe the difference in run-time for both the algorithms. As e-NFIS algorithm finds only NFIS, we consider running time only for finding NFIS for both the algorithms for a fair comparison. The results have been displayed in **Figures 4-6**.

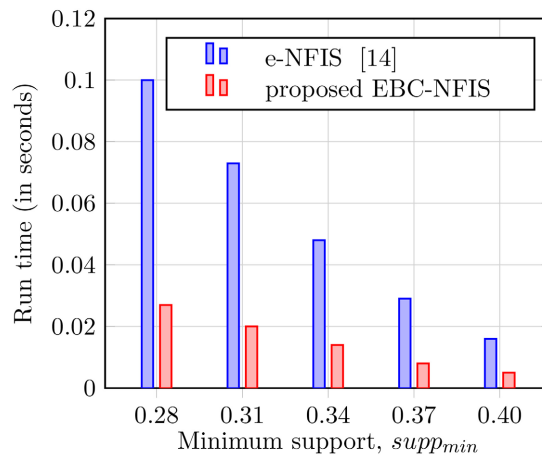
The running time shows that the proposed backward calculation method requires significantly less running time than that of e-NFIS. As minimum support decreases, the difference becomes more prominent. It is evident from the result that proposed method can discover NFIS far more efficiently than e-NFIS algorithm and consumes much less time. The difference in performance becomes even more noticeable when the number of possible frequent patterns increases. Bigger amount of patterns is usually associated with more distinct items and therefore number of negative items in an itemset.



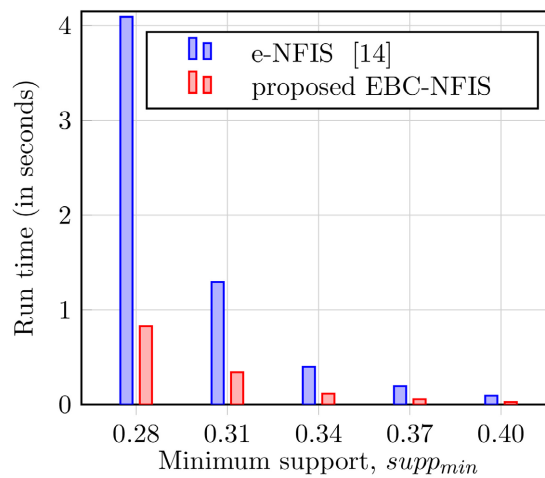
**Figure 2.** Positive and negative association rules for different  $supp_{min}$  and  $conf_{min}$  for mushroom dataset.



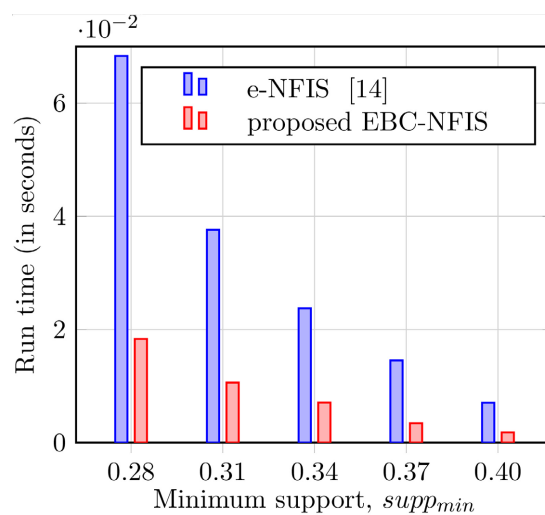
**Figure 3.** Positive and negative association rules for different  $supp_{min}$  and  $conf_{min}$  for congressional voting 84 dataset.



**Figure 4.** Comparison of runtime between e-NFIS and EBC-NFIS for different minimum supports for breast cancer dataset.



**Figure 5.** Comparison of runtime between e-NFIS and proposed EBC-NFIS for different minimum supports for mushroom dataset.



**Figure 6.** Comparison of runtime between e-NFIS and proposed EBC-NFIS for different minimum supports for congressional voting 84 dataset.

## 5. Conclusion

In this paper, we proposed an algorithm EBC-NFIS to discover PNARs and NFIS from positive ones. The algorithm is based on popular algorithm called e-NFIS. The proposed EBC-NFIS algorithm mines positive frequent itemsets using FP-Growth algorithm and generates candidate negative itemsets. Candidate negative itemsets are filtered based on a support threshold. We utilize calculated values of previous step to speed up calculation of current step. Experimental results showed that EBC-NFIS calculates support exponentially faster than e-NFIS which validate efficiency of our proposal. However, EBC-NFIS consumes higher memory compared to e-NFIS. Reduction in memory consumption will be our future research.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Mahmood, S., Shahbaz, M. and Guergachi, A. (2014) Negative and Positive Association Rules Mining from Text Using Frequent and Infrequent Itemsets. *The Scientific World Journal*, **2014**, Article ID: 973750. <https://doi.org/10.1155/2014/973750>
- [2] Çokpınar, S. and Gündem, T.I. (2012) Positive and Negative Association Rule Mining on XML Data Streams in Database as a Service Concept. *Expert Systems with Applications*, **39**, 7503-7511. <https://doi.org/10.1016/j.eswa.2012.01.128>
- [3] Kadir, A.S.A., Bakar, A.A. and Hamdan, A.R. (2011) Frequent Absence and Presence Itemset for Negative Association Rule Mining. *International Conference on Intelligent Systems Design and Applications*, Cordoba, 22-24 November 2011, 965-970. <https://doi.org/10.1109/ISDA.2011.6121783>
- [4] Folasire, O., Akinyemi, O. and Owoaje, E. (2014) Perceived Social Support among HIV Positive and HIV Negative People in Ibadan, Nigeria. *World Journal of AIDS*, **4**, 15-26. <https://doi.org/10.4236/wja.2014.41003>
- [5] Agrawal, R. and Srikant, R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. *VLDB94: Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago de Chile, 12-15 September 1994, 487-499.
- [6] Han, J., Pei, J. and Yin, Y. (2000) Mining Frequent Patterns without Candidate Generation. *ACM Sigmod Record*, **29**, 1-12. <https://doi.org/10.1145/335191.335372>
- [7] Brin, S., Motwani, R. and Silverstein, C. (1997) Beyond Market Baskets: Generalizing Association Rules to Correlations. *ACM Sigmod International Conference on Management of Data*, Tucson, 13-15 May 1997, 265-276. <https://doi.org/10.1145/253262.253327>
- [8] Savasere, A., Omiecinski, E. and Navathe, S. (1998) Mining for Strong Negative Associations in a Large Database of Customer Transactions. *Proceedings 14th International Conference on Data Engineering*, Orlando, 23-27 February 1998, 494-502.
- [9] Wu, X., Zhang, C. and Zhang, S. (2004) Efficient Mining of both Positive and Negative Association Rules. *ACM Transactions on Information Systems (TOIS)*, **22**, 381-405. <https://doi.org/10.1145/1010614.1010616>
- [10] Cornelis, C., Yan, P., Zhang, X. and Chen, G. (2006) Mining Positive and Negative

- Association Rules from Large Databases. *IEEE Conference on Cybernetics and Intelligent Systems*, Bangkok, 7-9 June 2006, 1-6.  
<https://doi.org/10.1109/ICCIS.2006.252251>
- [11] Dong, X., Sun, F., Han, X. and Hou, R. (2006) Study of Positive and Negative Association Rules Based on Multi-Confidence and Chi-Squared Test. *International Conference on Advanced Data Mining and Applications*, Xi'an, 14-16 August 2006, 100-109. [https://doi.org/10.1007/11811305\\_10](https://doi.org/10.1007/11811305_10)
- [12] Dong, X., Niu, Z., Shi, X., Zhang, X. and Zhu, D. (2007) Mining both Positive and Negative Association Rules from Frequent and Infrequent Itemsets. *International Conference on Advanced Data Mining and Applications*, Harbin, 6-8 August 2007, 122-133. [https://doi.org/10.1007/978-3-540-73871-8\\_13](https://doi.org/10.1007/978-3-540-73871-8_13)
- [13] Antonie, M.L. and Zaane, O.R. (2004) Mining Positive and Negative Association Rules: An Approach for Confined Rules. *European Conference on Principles of Data Mining and Knowledge Discovery*, Pisa, 20-24 September 2004, 27-38.  
[https://doi.org/10.1007/978-3-540-30116-5\\_6](https://doi.org/10.1007/978-3-540-30116-5_6)
- [14] Dong, X., Ma, L. and Han, X. (2011) E-NFIS: Efficient Negative Frequent Itemsets Mining Only Based on Positive Ones. *IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, 27-29 May 2011, 517-519.
- [15] Das, A., Ng, W.K. and Woon, Y.K. (2001) Rapid Association Rule Mining. *Proceedings of the 10th International Conference on Information and Knowledge Management*, Atlanta, 5-10 October 2001, 474-481.
- [16] Zaki, M.J. (2000) Scalable Algorithms for Association Mining. *IEEE Transactions on Knowledge and Data Engineering*, **12**, 372-390.
- [17] Li, X., Liu, Y., Peng, J. and Wu, Z. (2002) The Extended Association Rules and Atom Association Rules. *Journal of Computer Research and Application*, **12**, 1740-1750.
- [18] Asuncion, A. and Newman, D. (2013) UCI Machine Learning Repository. School of Information and Computer Science, University of California, Irvine, 28.  
<https://archive.ics.uci.edu/ml/index.php>