VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Le Tuan Anh

OPTIMIZE CNF ENCODING FOR ITEMSET MINING TASKS

Major: Computer Science

HA NOI - 2024

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Le Tuan Anh

OPTIMIZE CNF ENCODING FOR ITEMSET MINING TASKS

Major: Computer Science

Supervisor: Dr. To Van Khanh

HA NOI - 2024

ABSTRACT

Summary: In this thesis, we apply the "Sequential Encounter Encoding" method to optimize itemset mining tasks. This method has been proven effective in optimizing the search process for itemsets in data. By utilizing "Sequential Encounter Encoding" we aim to enhance the performance of itemset mining algorithms while minimizing processing time.

We conduct a series of experiments on real-world datasets to evaluate the performance of the applied method. Experimental results demonstrate a significant improvement in accuracy and efficiency compared to traditional methods. The application of this method not only enhances the performance of data mining processes but also opens up potential applications for similar problems in the field of data science and information retrieval.

Keywords: *SAT, SAT Encoding, Sequential Encounter Encoding, Itemset Mining Tasks*

ACKNOWLEDGEMENT

First of all, I would like to express my deepest gratitude to Lecturer, Dr. To Van Khanh, who wholeheartedly guided, guided, encouraged, and helped me throughout the course of this thesis.

I would like to thank the teachers in the Faculty of Information Technology as well as the University of Engineering and Technology - Vietnam National University, Hanoi, for creating conditions for me to study in a good environment and for teaching me the best things to do. Knowledge is especially important for me to continue to study and work in the future.

I especially express my gratitude to my family who have always been a solid support to help and support me in every journey. Finally, I would also like to thank the members of the K65CA-CLC2 class, my friends in the course, and the siblings inside and outside the school who have been with me to study and practice and help each other throughout four years university.

Thank you sincerely!

Le Tuan Anh

AUTHORSHIP

I hereby declare that the thesis ”OPTIMIZE CNF ENCODING FOR ITEMSET MINING TASKS” is done by me and has never been submitted as a report for Graduation Thesis at University of Engineering and Technology - Vietnam National University, Hanoi, or any other university. All content in this thesis is written by me and has not been copied from any source, nor is the work of others used without specific citation. I also warrant that the source code is my development and does not copy the source code of any other person. If wrong, I would like to take full responsibility according to the regulations of University of Engineering and Technology - Vietnam National University, Hanoi.

Ha Noi, May 26 2024

Student

Le Tuan Anh

SUPERVISOR'S APPROVAL

I hereby approve that the thesis in its current form is ready for committee examination as a requirement for the Bachelor of Computer Science degree at the University of Engineering and Technology.

Ha Noi, May 26 2024

Supervisor

Dr. To Van Khanh

Contents

List of Figures

CHAPTER 1

Introduction

This chapter will focuses on introducing the itemset mining tasks and SAT encoding, encompassing their concepts, related terms, and applications.

1.1 Itemset Mining Tasks

1.1.1 Overview

Frequent item sets are a key technique in the realm of data mining, specifically aimed at uncovering relationships among different items within a dataset. The essence of association rule mining lies in identifying those item relationships that occur frequently together in the dataset.

In simpler terms, a frequent item set refers to a collection of items that commonly appear together in the dataset. We measure the frequency of an item set using what's known as the support count. This count tells us how many times the particular set of items crops up together in transactions or records within the dataset. In practice, the goal is to find item sets with a minimum support, indicating how frequently they occur in transactions or records within the dataset.

For example, with a dataset of transactions from a retail store

Tid	**Itemsets**
1	apple, banana, cherry
2	apple, mango
3	apple, cherry
4	mango, cherry
5	apple, mango, cherry

Table 1.1: Example of a dataset of transactions

With minimum support is 3, we need to find all itemsets appearing in at least 3 transactions and return the following result:

- Itemset 1: {apple, cherry} in transactions [1, 3, 5]
- Itemset 2: {apple} in transactions [1, 2, 3, 5]

1.1.2 Technical background

Firstly, we establish several symbols to represent the itemset mining problem. These symbols aid in formalizing the problem and defining key concepts. For instance, we denote:

- Ω: a set of all items
- I: an itemset in Ω, where $I \subseteq \Omega$
- T_i: a transaction identifier. For $T_i = (i, I)$
- D: a transaction database, where D contains a set of transactions, $D = \{T_1, T_2, ..., T_n\}$
- $Supp(I, D)$: the support of itemset I in database D, where $Supp(I, D)$ is the number of transactions in D that contain I

For example, in table 1.1, we have:

- Ω is {apple, banana, cherry, mango}
- I can be {apple}, {apple, mango}, {apple, mango, cherry}, ...
- D = {(1, {apple, banana, cherry}), (2, {apple, mango}), (3, {apple, cherry}), (4, {mango, cherry}), (5, {apple, mango, cherry})}
- T_1 = (1, {apple, banana, cherry}), T_2 = (2, {apple, mango}), T_3 = (3, {apple, cherry}),...
- $Supp(\{apple, cherry\}, D) = 3$, $Supp(\{apple\}, D) = 4$,...

Let λ be the minimum support threshold, the frequent itemset mining problem is to find all itemsets I such that $Supp(I, D) \geq minsup$. In general, it can present by:

$$FIM(D, \lambda) = \{\mathrm{I} \subseteq \Omega \mid Supp(I, D) \geq \lambda\}$$

1.2 SAT Encoding

1.2.1 Concept

1.2.2 SAT Solvers

1.2.3 Applications

CHAPTER 2

SAT-based Encoding of Itemset Mining

In this chapter, we present how to encode the itemset mining problem into a SAT problem. And we will discuss the limitation of the standard method.

2.1 Constraint Encoding

To resolve the itemset mining problem, we using the SAT encoding approach. In essence, SAT encoding involves the creation of variables and the imposition of constraints to represent the itemset mining problem. These variables serve to denote the presence or absence of items within a candidate itemset and are subjected to linear inequalities to ensure the itemset's support.

In the context of a transaction database $D = (1,T_1),...,(\mathrm{m},T_m)$ and a minimum support threshold λ, each item in the candidate itemset X, we denote:

- p_a: is *true* if the item a is in the itemset X, otherwise $p_a = false$
- q_i: is *true* if the transaction T_i contains the itemset X, otherwise $q_i = false$

Alongside, a set of constraints is imposed on these variables to establish a one-to-one correspondence between the models of the resulting CNF formula and the set of itemsets.

Firstly, to capture all the transactions where the candidate itemset does not appear, we use following constraint:

$$\bigwedge_{i=1}^{m} (q_i \leftrightarrow \bigwedge_{a \notin T_i} \neg p_a) \tag{2.1}$$

This constraint guarantees that q_i is true if and only if either all items not in T_i are also not in the itemset X, or transaction T_i contains the itemset X.

Constraint 2.1 can be rewritten as follows:

$$\bigwedge_{a \in \Omega} \bigwedge_{a \notin T_i} (\neg p_a \vee \neg q_i) \tag{2.2}$$

$$\bigwedge_{T_i \in D} ((\bigvee_{a \notin T_i} p_a) \vee q_i) \tag{2.3}$$

Finally, the frequency constraint, can be simply expressed as follows:

$$\sum_{i=1}^{m} q_i \geq \lambda \tag{2.4}$$

For example, with a dataset of transactions from a retail store in table 1.1, we mark: a = apple, b = banana, c = cherry, d = mango. Then we have database transactions

Tid	**a**	**b**	**c**	**d**
1	1	1	1	0
2	1	0	0	1
3	1	0	1	0
4	0	0	1	1
5	1	0	1	1

Table 2.1: Example of a dataset of transactions after convert

The itemset mining problem will be defined as:

$$q_1 \leftrightarrow (\neg p_d)$$
$$q_2 \leftrightarrow (\neg p_b \wedge \neg p_c)$$
$$q_3 \leftrightarrow (\neg p_b \wedge \neg p_d)$$
$$q_4 \leftrightarrow (\neg p_a \wedge \neg p_d)$$
$$q_5 \leftrightarrow (\neg p_b)$$
$$q_1 + q_2 + q_3 + q_4 + q_5 \geq \lambda$$

In the next step, we must encode constraint 2.4 into CNF formula.

2.2 Standard Method in Itemset Mining

After encoding the base constraints, we proceed to employ the standard method to encode formula 2.4.

To solve the problem $q_1 + q_2 + ... + q_n \geq \lambda$, we can use the standard method known as C_{n-k+1}.

The algorithm's idea is as follows: Suppose we have a set of n elements. If there are at least k true elements, it is equivalent to having at most $n - k$ false elements. In other words, when selecting $n - k + 1$ elements, we are guaranteed to have at least one true element among them.

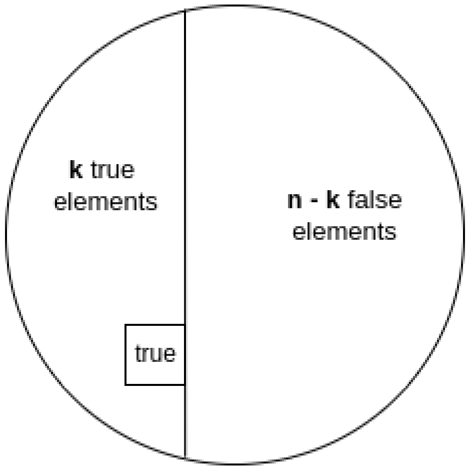

Figure 2.1: Illustration of the standard method C_{n-k+1}

For example, suppose $n = 5$ and $\lambda = 3$, we can use the constraint below to represent the concept of having at least 3 true elements among 5 elements.

$$\begin{aligned}
&(q_1 \vee q_2 \vee q_3) \\
\wedge&(q_1 \vee q_2 \vee q_4) \\
\wedge&(q_1 \vee q_2 \vee q_5) \\
\wedge&(q_1 \vee q_3 \vee q_4) \\
\wedge&(q_1 \vee q_3 \vee q_5) \\
\wedge&(q_1 \vee q_4 \vee q_5) \\
\wedge&(q_2 \vee q_3 \vee q_4) \\
\wedge&(q_2 \vee q_3 \vee q_5) \\
\wedge&(q_2 \vee q_4 \vee q_5) \\
\wedge&(q_3 \vee q_4 \vee q_5)
\end{aligned} \tag{2.5}$$

Then with n elements and λ, we can present the constraint as:

$$\bigwedge_{i=1}^{n-\lambda+1} \left(\bigvee_{j=i}^{i+\lambda-1} q_j \right) \tag{2.6}$$

Now, we can use constraints 2.2, 2.3 and 2.6 to resolve the problem Itemset Mining.

2.3 Limitation of Standard Method

The standard method C_{n-k+1} is a widely used approach to solve various problems, including itemset mining. However, its major drawback lies in the explosion of combinations, especially when k approaches $n/2+1$.

In a simple example, with $n = 30$ and $\lambda = 16$, the number of clauses required to encode C_{15}^{30} is approximately 155 million. In reality, n corresponds to the number of transactions, which can range from thousands to millions. Therefore, the standard method is not feasible for large datasets.

CHAPTER 3

Sequential Encounter Encoding for Optimization

3.1 Sequential Encounter Encoding

3.2 Optimization CNF Encoding using Sequential Encounter Encoding

First, we add following constraint for relationship between q and r:

$$\begin{aligned} q_i \rightarrow r_{i1} \quad \forall i \in [1, n] \\ \neg q_i \rightarrow \neg r_{ii} \quad \forall i \in [1, n] \end{aligned} \tag{3.1}$$

This constraint guarantees that if q_i is true, then r_{i1} must be true. And if q_i is false, r_{ii} must be false.

Secondly, we encode

$$\neg r_{ij} \quad \forall i \in [1, \lambda - 1], j \in [i + 1, \lambda] \tag{3.2}$$

Thirdly, we encode

$$\neg r_{i-1,j} \rightarrow r_{ij} \quad \forall i \in [2, n], j \in [1, \lambda] \tag{3.3}$$

Fourthly, we encode

$$\begin{aligned} q_i \wedge r_{i-1,j-1} \rightarrow r_{ij} \quad \forall i \in [2, n], j \in [2, \lambda] \\ \neg q_i \wedge \neg r_{i-1,j} \rightarrow \neg r_{ij} \quad \forall i \in [2, n], j \in [1, \lambda] \\ \neg r_{i-1,j} \wedge \neg r_{i-1,j-1} \rightarrow \neg r_{ij} \quad \forall i \in [2, n], j \in [2, \lambda] \end{aligned} \tag{3.4}$$

Finally, we encode

$$\begin{gathered}
r_{n-1,\lambda} \vee (q_\lambda \wedge r_{n-1,\lambda-1}) \\
\neg q_n \rightarrow r_{n-1,\lambda} \\
\neg q_i \wedge r_{j,\lambda} \rightarrow r_{i-1,\lambda} \quad \forall i \in [\lambda+1, n]
\end{gathered} \tag{3.5}$$

CHAPTER 4

Experiments

4.1 Experimental Setup and Datasets

4.2 Results and Analysis

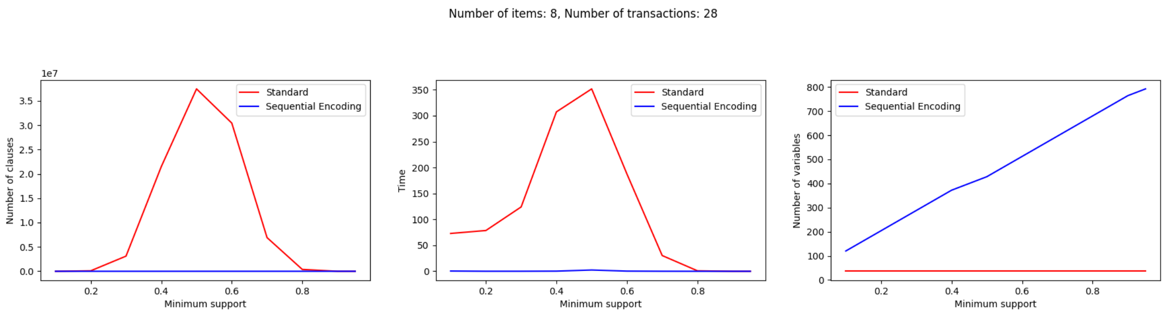

Figure 4.1: Comparison of the number of clauses and the number of variables in the CNF encoding of the optimization problem using the sequential encounter encoding and the direct encoding.

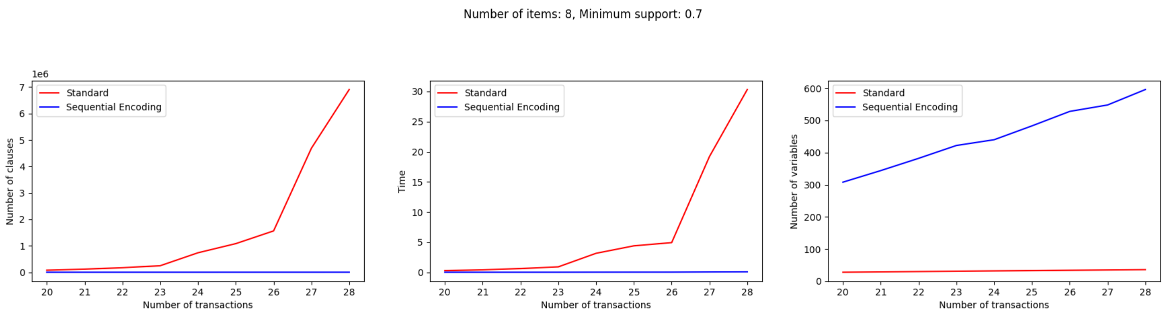

Figure 4.2: Comparison of the number of clauses and the number of variables in the CNF encoding of the optimization problem using the sequential encounter encoding and the direct encoding.

Conclusions