Occupancy Based Pattern Mining: Current Status And Future Directions

Jhimli Adhikari
Narayan Zantye College, Goa, India

The main purpose of data mining and analytics is to find novel, potentially useful patterns that can be utilized in real-world applications to derive beneficial knowledge. In recent years, a new measure of pattern interestingness called occupancy of a pattern was introduced to ensure that each pattern found represents a large part of transactions where it appears. Main objective of this measure is to enhance the quality of a pattern. This article surveys recent studies on pattern mining and its applications based on occupancy. The goal of the paper is to provide both an introduction to occupancy based pattern mining (OPM), and a survey of recent advances and research opportunities. Moreover, main approaches and strategies to solve occupancy based pattern mining problems are also presented. The paper also presents challenges and research opportunities of using occupancy measure in other popular pattern mining problems.

Keywords: Database, occupancy, pattern mining, task oriented pattern, utility.

1. INTRODUCTION

In many real-world applications, data mining techniques are used to extract interesting patterns from databases, to support crucial decision-making. Among the various types of data mining applications, analysis of transactional data has been considered important. Two fundamental tasks for revealing interesting relationships between items in transactional databases are frequent itemset mining (FIM) and association rule mining (ARM) (Agrawal, Imielinski, and Swami, 1993; M.S.Chen, Han, and P.S.Yu, 1996; P.F.Viger, P. Lin, Vo, and Le, 2017). FIM and ARM are considered as basic pattern mining and this is one of the important and fundamental data mining techniques to extract meaningful and useful information from massive amount of data (Moens, Aksehirli, and Goethals, 2013). Here support (FIM) and confidence (ARM) are the key measures for patterns interestingness. Several patterns such as calendar based pattern (Adhikari, 2014), utility based pattern (Gan, Lin, F.Viger, Chao, Tseng, and Yu, 2018), temporal patterns (Wang, Meng, Xu, and Peng, 2018) etc. were derived from these basic patterns in recent decade. Recently, a new measure of pattern interestingness called occupancy of a pattern was introduced by (Tang, Zhang, Luo, and Wang, 2012) to ensure that each pattern found represents a large part of transactions where it appears.
This measure ensures that any interesting pattern should occupy a large portion of the transactions it appears in. Thus, it enhances the quality of a pattern. It can be seen as the relative size of a pattern to its supporting transactions (sequences) rather than the absolute size of it, and might be more meaningful in many applications. With the definition of occupancy, a pattern is called dominant if its occupancy value crosses user specified threshold. In the following section we explain how occupancy measure is used to determine dominant pattern.

1.1 Occupancy in frequent pattern mining

Frequent itemset mining is a conventional problem in data mining. An itemset is called frequent if its support is not less than users defined threshold. Occupancy measure is used to mine a pattern which occupies large portion of the transactions. With the definition of occupancy a pattern is called dominant if its occupancy is above a user-specified threshold. The support of a recommended pattern correlates to the recommendation precision while its occupancy is related to the recommendation recall. With the definition of support and occupancy we can measure the

quality of an itemset by combining these two factors. In this connection an algorithm DOFIA (DOminant and Frequent Itemset mining Algorithm) is proposed by (Tang et al., 2012) to mine the qualified dominant patterns from transaction database. Another form of occupancy was introduced to reflect the interestingness of desired pattern called weight occupancy (Gan, Lin, F.Viger, Chao, Zhan, and Zhang, 2018). The weight occupancy can lead to useful itemsets that contribute a large portion of total weight for each individual transaction representing user interests or user habit. In this regard, a tree based algorithm called exploiting highly Qualified patterns with Frequency and Weight Occupancy (QFWO), was developed to suggest the possible highly qualified patterns that utilize the idea of co-occurrence and weight occupancy. Recently, (Deng, 2020) proposed an algorithm HEP (abbreviation for High Efficient algorithm for mining high occupancy itemsets), to discover all high occupancy itemsets. The algorithm uses a data structure, named occupancy-list, to store the occupancy information about an itemset.

1.2 Occupancy in Utility based pattern mining

Frequency-based FIM framework often leads to many patterns which are not actionable at all (Cao, 2013). To enhance pattern action ability, one effort is to extract high utility pattern which considers the utilities of unit items and itemsets in addition to their statistical significance. The existing FIM and association rule mining algorithms are incapable of capturing such high utility patterns, because they ignore the business interest of each item and itemset which is essential for decision-making (Cao, Yu, Zhang, and Zhao, 2010). Thus, High Utility Itemset (HUI) Mining emerges as a recent solution for enhancing pattern action ability since it was first introduced in 2004 (Yao, Hamilton, and Butz, 2004). Recently concept of utility occupancy is adopted to evaluate the utility contribution of patterns in their supporting transactions. Compared with occupancy, utility occupancy is suitable and more effective for pattern analysis in some real-life domains.

1.3 Occupancy in Task oriented pattern mining

Objectives of task oriented pattern mining is to mine the most frequent and complete pattern (i.e., set of items) in a transaction dataset for recommendation. Applications of task oriented pattern mining are increasing in real life problems. Here support measure is used to capture the popularity of patterns, while occupancy measure is adopted to capture the completeness of patterns. Due to the conflicting nature between support and occupancy, task-oriented pattern mining can be regarded as a multi-objective optimization problem (MOP). (Zhang, Duan, Zhang, Cheng, Jin, and Tang, 2017) proposed an evolutionary algorithm called (Multi-Objective Pattern Mining algorithm) to solve the task-oriented pattern mining problem from a multi objective perspective, which can strike a balance between support and occupancy.

2. MOTIVATION OF THE STUDY

The number of patterns generated during the Data Mining process is very large but only a few of these patterns are likely to be of any interest to the domain expert analyzing the data. Although, support (frequency) measures a patterns' interestingness, many of the patterns are either irrelevant or obvious and do not provide new knowledge. The selection of the most representative patterns for a data set is another important issue in terms of the quality assessment. In the previous section we have seen that how occupancy measure is incorporated in mining frequent pattern and utility based patterns. It ensures to find qualified patterns. To measure the utility of a pattern, a utility-based mining framework called high utility pattern mining (HUPM) was first proposed by (Yao, Hamilton, and Butz, 2004). Recently, (Shen, Wen, Zhao, Zhou, and Zheng, 2016) proposed an algorithm called OCEAN to address the problem of high utility occupancy pattern mining by introducing the utility occupancy measure. Concept of occupancy in sequential pattern was introduced in algorithm DOFRA by Zhang et al. (2015). (Gan, Lin, F.Viger, Chao, and Yu, 2020) incorporated occupancy measure to mine high utility patterns and proposed HUOPM (High Utility Occupancy Pattern Mining) to extract high quality patterns from

weighted transactions. Extracting patterns that occupy a large portion of the utility in their supporting transactions is useful in several applications. Thus, one could see the occupancy based mining is gaining popularity slowly to mine quality patterns as patterns with higher occupancy may lead to higher recall while patterns with higher frequency lead to higher precision. Thus, development of occupancy-oriented Pattern mining algorithm (OPM) has been an important issue in data-mining research. The major contributions of this paper are as follows

(a) **T**his paper presents a comprehensive survey of OPM (Occupancy based Pattern Mining). This survey investigates relevant papers published in the last 15 years and summarizes them in a systematic fashion.
(b) **T**his survey deeply and comprehensively summarizes the developments of this field, comparing state-of-the-art work to earlier work. It introduces an in-depth understanding of occupancy measure used in different types of databases.
(c) **A** comprehensive review of existing algorithms is presented, with an in-depth discussion of their pros and cons.
(d) **W**e further provide an in-depth summary and discussion on the characteristics of the current algorithms.
(e) **F**inally, we identify several important issues and research opportunities for occupancy based pattern mining.

The rest of the paper is organized as follows. In Section 3, we introduce the necessary background information, basic concepts and example. Section 4 and 5 introduce various occupancy based pattern mining algorithms. Section 6 identifies key issues and research opportunities related to occupancy based patterns. Finally, Section 7 draws a conclusion.

3. PRELIMINARIES

This section introduces important preliminaries and notations. A transaction database is a set of transactions, where each transaction is a set of items. Let I be the complete set of distinct items and T be the complete set of transactions. Any non-empty set of items is called an itemset and any set of transactions is called a transaction set. The transactions that contain all the items in an itemset X are the supporting transactions of X, denoted as Tx. The frequency of an itemset X is the number of transactions in Tx. The support and frequent itemset concepts were introduced by (Agrawal, Imielinski, and Swami, 1993). (Tang, Zhang, Luo, and Wang, 2012) first introduced the occupancy measure in binary transaction database. In the following subsections we introduce the concepts of occupancy measure in various types of databases.

3.1 Occupancy measure in transaction database

Definition 1 (Occupancy): The occupancy of an itemset X is defined as follows

$$\phi(X) = avg(\{\frac{|X|}{|t|} : t \in T_x\})$$

where avg() is the average function of all the values in the set. Two different average functions, harmonic average and arithmetic average, could be used in the above definition.

Definition 2 (Harmonic occupancy): The harmonic occupancy of an itemset X is defined as follows

$$\phi_H(X) = HAvg(\{\frac{|X|}{|t|} : t \in T_x\}) = \frac{|T_x||X|}{\sum_{t \in T_x} |t|}$$

Where

$$HAvg(X) = \frac{|X|}{\sum_{x \in X} \frac{1}{x}}$$

is the harmonic average of a set of numbers X. From above it is clear that the occupancy of an itemset X is the average ratio of the occurrences of the items in X to the number of the items in the transaction it appears in. Therefore, the high value of the occupancy indicates that besides the items in X there are only a small number of items left inside the supporting transactions of X.

Definition 3 (Arithmetic occupancy): The arithmetic occupancy is defined as

$$\phi_A(X) = AAvg(\{\frac{|X|}{|t|} : t \in T_x\}) = \frac{1}{|T_x|}\sum \frac{|X|}{|t|}$$

Example 1: Consider the following transaction database D_1 shown in Table I where TID is the transaction identifier to elaborate the definition of occupancy. Database consists of 10 transactions.

Table I: A transaction database D_1

TID	Items purchased	Length	TID	Items purchased	Length
t_1	1, 2, 4, 7, 8, 9, 10, 14, 15, 16, 20, 21	12	t_6	1, 2, 5, 6, 7, 9, 12, 14, 15, 17, 19, 21	12
t_2	2, 5, 7, 9, 12, 14, 15, 20	8	t_7	2 ,7, 14, 20	4
t_3	2, 7, 13, 14, 20	5	t_8	2, 7, 14, 20	4
t_4	1, 2, 4, 5, 7, 8, 14, 15, 18, 20, 21	11	t_9	2, 7, 14, 20	4
t_5	2, 3, 7, 11, 14, 21	6	t_{10}	2, 14, 20	3

Consider two itemsets $I_1 = \{2, 7, 14, 20\}$ and $I_2 = \{2, 7, 14, 15, 20\}$ in Table I. The supporting transactions of I_1 and I_2 are $\{t_1, t_2, t_3, t_4, t_7, t_8, t_9\}$ and $\{t_1, t_2, t_4\}$ respectively. We calculate the harmonic occupancy of the two itemsets using Definition 1.

$$\phi_H(I_1) = \frac{4 \times 7}{12 + 8 + 5 + 11 + 4 \times 3} \approx 0.54$$

$$\phi_H(I_2) = \frac{5 \times 3}{12 + 8 + 11} \approx 0.48$$

From above example it is seen that although $I_1 \subset I_2$, we have $\phi_H(I_1) > \phi_H(I_2)$ The reason is that I_2 only appears in large transactions where it only occupies a small fraction, while I_1 appears in many smaller transactions where it occupies a large fraction. Thus, occupancy does not always increase monotonically when more items are added to an itemset. Similarly, occupancy does not always decrease monotonically when we add more items to an itemset either.

3.2 Occupancy measure in sequence database

A sequence database consists of sequences of ordered elements or events, recorded with or without a concrete notion of time (Agrawal and Srikant, 1995). The task of sequential pattern mining is a data mining task specialized for analyzing sequential data, to discover sequential patterns. More precisely, it consists of discovering interesting subsequences in a set of sequences, where the interestingness of a subsequence can be measured in terms of various criteria such as its occurrence frequency, length, and profit.

Let I be the complete set of distinct items. A sequence database is a set of sequences, in which each sequence is an ordered list of events which is denoted as $e_1e_2...e_m$. If each event in a given sequence is an item, we call it single-itemset sequence. Otherwise, we call it multiple-itemset sequence in which each event is an itemset. In this article, we only focus on single-itemset sequences since it represents one of the most important and popular type of sequences, such as protein sequences, DNA strings, Web click streams and the travel-package sequences.

We use the pattern P to represent a subsequence and the database D to represent a sequence

database. The number of items in P, denoted by |P|, is called the length of P and a pattern with a length l is called a l-pattern. The sequences in D that contains a pattern P are the supporting sequences of P, in database D denoted as D_P. The frequency of a pattern P in D is defined as $freq(P) = |D_P|$. The support of a pattern P is the percentage of sequences in D that contains P which is defined as $supp(P) = \frac{freq(P)}{|D|}$. For a given minimum support threshold $\alpha(0 < \alpha \leq 1)$, a pattern P is said to be frequent if $supp(P) \geq \alpha$.

Example 2: Consider the following sequence database D_2 shown in Table II where Sid is the sequence identifier. This database has nine unique items (a, b, c, d, e, f, g, h, i) and five input sequences ($|D_2| = 5$). Suppose P = abc, there are four sequences (S_1, S_2, S_3, S_4) containing this pattern, thus freq(P) = 4 and supp(P) = 4/5 = 0.8. P is frequent if $\alpha = 0.5$.

Table II: A sequence database D2

Sid	**Sequence**	**Length**
S_1	a b c d e g	6
S_2	a b c f	4
S_3	a b c	3
S_4	a b c	3
S_5	a b e h i	5

Definition 4 (Occupancy): Formally, the occupancy of pattern P is defined as $\phi(P) = AVG(\{\frac{|P|}{|S|} : S \in Dp\})$ where AVG() is the average function of all the values in the set. Two different average functions, harmonic average and arithmetic average, could be used in the above definition. One could use harmonic occupancy / arithmetic occupancy which is nothing but average functions (i.e., arithmetic average and harmonic average) depending on applications. It also depends on type of data i.e., the arithmetic average is best used in situations in which the data are not skewed (no extreme outliers). One could use the harmonic average when there is: a large population in which the majority of the values are distributed uniformly but where there are a few outliers with significantly higher values (Haff, 1979).
Consider Example 2, to calculate the occupancy of the following three patterns P_1 = ab, P_2 = abc, and P_3 = abcd. The supporting sequences of patterns P_1, P_2, and P_3 are $\{S_1, S_2, S_3, S_4, S_5\}$, $\{S_1, S_2, S_3, S_4\}$, and $\{S_1\}$ respectively. Then harmonic occupancy $\phi_H(P)$ of patterns are computed as follows

$$\phi_H(P_1) = \frac{5 \times 2}{6+4+3+3+5} \approx 0.48$$

$$\phi_H(P_2) = \frac{4 \times 3}{6+4+3+3} \approx 0.75$$

$$\phi_H(P_3) = \frac{4}{6} \approx 0.67$$

The arithmetic occupancy $\phi_A(P)$ of the patterns are computed as follows

$$\phi_A(P_1) = \frac{\frac{2}{6}+\frac{2}{4}+\frac{2}{3}+\frac{2}{3}+\frac{2}{5}}{5} \approx 0.51$$

$$\phi_A(P_2) = \frac{\frac{3}{6}+\frac{3}{4}+\frac{3}{3}+\frac{3}{3}}{4} \approx 0.81$$

$$\phi_A(P_3) = \frac{4}{6} \approx 0.67$$

Lemma: If the pattern appears in one transaction then occupancy value of harmonic occupancy and arithmetic occupancy value is same. This occupancy measure helps to find dominant patterns from the database.

Definition 5 (Dominant Pattern): For a given minimum occupancy threshold $\beta(0 < \beta \leq 1)$, the pattern P is said to be dominant if $\phi(P) \geq \beta$. The quality of a pattern could be measured by combining two factors support and occupancy.

Definition 6 (Quality Pattern): The quality (value) of a pattern P is defined as $q(P) = Q(supp(P), \phi(P))$ where $Q(supp(P), \phi(P))$ is any function, which maps the support and occupancy to a real value.

One could use the weighted sum function to find quality pattern, that is, $q(P) = supp(P) + \lambda\phi(P)$, where the weight $\lambda(0 \leq \lambda < +\infty)$ is a user defined parameter to capture the relative importance of support and occupancy. It is worth mentioning that any other functions, such as the harmonic average (similar to the F1 score) and the sum of logarithms (similar to block size proposed in (Gade, Wang, and Karypis, 2004) can be used to combine the two values of support and occupancy. Later, we will see that the proposed techniques can be applied to any function $Q(supp(P), \phi(P))$, which is monotonically increasing with respect to $\phi(P)$.

Definition 7 (Qualified Itemset): For a given minimum support threshold α and a minimum occupancy threshold $\beta(0 < \alpha, \beta \leq 1)$, pattern P is said to be qualified if $supp(P) \geq \alpha$ and $\phi(P) \geq \beta$.

3.3 Occupancy measure in weighted database

In real-world applications, each object may have a different importance to people. To address this issue, weighted frequent pattern mining concept was introduced.

Let $I = \{i_1, i_2, ..., i_m\}$ be a finite set of m distinct items in a transactional database $D = \{T_1, T_2, ..., T_n\}$, where each transaction $Tq \in D$ is a subset of I, and has an unique identifier Tid which is associated with a timestamp. In addition, each item i_m in D has a unique existence weight $w(i_m)$, provided in a weight table denoted as $wtable = \{w(i_1), w(i_2), ..., w(i_m)\}$. An itemset X is a set of k distinct items $\{i_1, i_2, ..., i_k\}(k \geq 1)$ and is called k-itemset. The user-specified minimum support threshold is denoted as α. The number of transactions containing an itemset is known as the occurrence frequency of that itemset and is called the support count of the itemset. The support count of an itemset X, denoted as supp(X), is the number of supporting transactions containing X w.r.t. $X \subseteq Tq$. An itemset, X, is designated as a frequent pattern in a database D if $supp(X) \geq \alpha \times |D|$.

Example 3: Consider a database D_3 consisting of 10 transactions and weight of 5 distinct items are shown in Table III and Table IV respectively.

Definition 8 The weight of an item i_j in a database is a value in the range of (0, 1], denoted as $w(i_j)$. It represents the importance of the item i_j according to users preferences.

Definition 9 The weight of an itemset X in a database D_3 is denoted as w(X) and defined as the sum of the weights of all items in X as:

$$w(X) = \sum_{i_j \in X} w(i_j)$$

The weights of the itemsets (b) and (bde) in the Example 3 are, respectively, calculated as w(b) = 0.75, and $w(bde) = w(b) + w(d) + w(e) = 0.75 + 0.5 + 0.3 = 1.55$.

Definition 10 The weight of an itemset X in a transaction T_q is equal to the weight of X in D_3 as: $w(X, T_q) = w(X)$

Definition 11 The weight of a transaction Tq is the total weight of items in T_q, that is:

$$tw(T_q) = \sum_{i_j \in T_q} w(i_j)$$

Table III: A weighted transaction database D_3

Tid	Transactions (item)	Tid	Transactions (item)
T_1	a c e	T_6	b c e
T_2	b d	T_7	b d
T_3	a b c	T_8	a b c d e
T_4	c e	T_9	d e
T_5	a c d e	T_{10}	b c e

Table IV: Weight of items

item	a	b	c	d	e
item weight(w)	0.2	0.75	1.0	0.5	0.3

where i_j is the j-item in T_q.

Definition 12 The weight occupancy of an itemset X in a transaction T_q is denoted as wo(X, Tq) and defined as the occupancy ratio of the weight values of the itemset X in this supporting transaction, that is:

$$wo(X, T_q) = \frac{w(X, T_q)}{tw(T_q)}$$

The weights of T_2 and T_3 in Example 3 are, respectively, calculated as $w(T_2) = w(b) + w(d) = 0.75 + 0.5 = 1.25$, and $w(T_3) = w(a) + w(b) + w(c) = 0.2 + 0.75 + 1.0 = 1.95$. The weight occupancy of (ab) in T_3 is calculated as $wo(ab, T_3) = (0.2 + 0.75)/(0.2 + 0.75 + 1.0) = 0.95/1.95 \approx 0.4872$, and the weight occupancy of (ab) in T_8 is calculated as $wo(ab, T_8) = (0.2 + 0.75)/(0.2 + 0.75 + 1.0 + 0.5 + 0.3) = 0.95/2.75 \approx 0.3455$.

Definition 13 The weight occupancy of an itemset X in a database D_3 is denoted as wo(X) and defined as:

$$wo(X) = \frac{\sum_{X \subseteq T_q \Lambda T_q \in D} wo(X, T_q)}{|\Gamma_X|}$$

where Γ_X is the set of supporting transactions of X in D_3 (thus $|\Gamma_X|$ is equal to the support of X in D_3).

Definition 14 Given a minimum support threshold $\alpha(0 < \alpha \leq 1)$ and a minimum weight occupancy threshold $\beta(0 < \beta \leq 1)$, an itemset X in a database D_3 is said to be a highly qualified pattern (HQP) with high frequency and strong weight occupancy, denoted as HQP, if it satisfies the following two conditions as: $sup(X) \geq \alpha \times |D_3|$ and $wo(X) \geq \beta$

The weight occupancies of (a) and (abc) in Table III are, respectively, calculated as: $wo(a) = (wo(a, T_1) + wo(a, T_3) + wo(a, T_5) + wo(a, T_8))/4 = (0.1333 + 0.1026 + 0.1000 + 0.0727)/4 \approx 0.1022, wo(abc) = (wo(abc, T_3) + wo(abc, T_8))/2 = (1.0000 + 0.709)/2 \approx 0.8545$. When α and β were set as 20% and 0.6, the complete set of HQPs of the database D_3 is shown in Table V. Clearly, the weight occupancy measure does not hold downward closure property.

Table V: High Quality Patterns in database D_3

itemset	ac	bc	bd	cd	ce	abc	acd	ace	bce	cde	acde
support	4	4	3	2	6	2	2	3	3	2	2
wo	0.6129	0.8103	0.8182	0.6477	0.7096	0.8545	0.7341	0.7652	0.9152	0.7773	0.8636

3.4 Occupancy measure in utility database

Let $I = \{i_1, i_2, ..., i_m\}$ be a finite set of m distinct items in a transactional quantitative database $D = \{T_1, T_2, ..., T_n\}$, where each quantitative transaction $T_q \in D$ is a subset of I, and has a unique identifier T_{id}. The total utility of all items in a transaction is named transaction utility and denoted as tu. An itemset X with k distinct items $\{i_1, i_2, ..., i_k\}$ is called a k-itemset.
Example 4: Let us consider a database D_4 consisting of 10 transactions shown in Table VI, which will be used as a running example for high utility patterns mining. Table VII shows 5 distinct items and their utility value.

Table VI: Transactional quantitative database D_4

tid	Transaction (item, quantity)	tu	tid	Transaction (item, quantity)	tu
T_1	a:2, c:4, d:7	\$65	T_6	c:2, e:4	\$58
T_2	b:2, c:3	\$37	T_7	c:2, d:1	\$23
T_3	a:3, b:2, c:1, d:2	\$38	T_8	a:3, b:1, d:2, e:4	\$61
T_4	b:4, d:3	\$11	T_9	a:2, c:4, d:1	\$59
T_5	a:1, b:3, c:2, d:5, e:1	\$49	T_{10}	c:3, e:1	\$42

Table VII: Profit of items

item	a	b	c	d	e
Unit profit	\$7	\$2	\$11	\$1	\$9

Definition 15: Each item i_m in a database D_4 has a unit profit, denoted as $pr(i_m)$, which represents its relative importance to the user. Item unit profits are indicated in a user-specified profit table, denoted as $ptable = \{pr(i_1), pr(i_2), ..., pr(i_m)\}$. The utility of an item i_j in a transaction T_q is defined as $u(i_j, T_q) = q(i_j, T_q) \times pr(i_j)$, in which $q(i_j, T_q)$ is the occur quantity, of i_j in T_q. The utility of an itemset/pattern X in a transaction T_q is defined as

$$u(X, T_q) = \sum_{i_j \in X \Lambda X \subseteq T_q} u(i_j, T_q)$$

. Thus, the total utility of X in a database D_4 is

$$u(X) = \sum_{X \subseteq T_q \Lambda T_q \in D} u(X, T_q)$$

Let us consider the itemsets (a) and (ab) in Example 3. Their utilities in T_3 are $u(a, T_3) = 3 \times \$7 = \$21$, and $u(ab, T_3) = 3 \times \$7 + 2 \times \$2 = \$21 + \$4 = \$25$, respectively. Thus, their utilities in the database are calculated as $u(a) = u(a, T_1) + u(a, T_3) + u(a, T_5) + u(a, T_8) + u(a, T_9) = \$14 + \$21 + \$7 + \$21 + \$14 = \$77$, and $u(ab) = u(ab, T_3) + u(ab, T_5) + u(ab, T_8) = \$21 + \$13 + \$23 = \$57$.
Definition 16: The transaction utility (tu) of a transaction T_q is

$$tu(T_q) = \sum_{i_j \in T_q} u(i_j, T_q)$$

where i_j is the j-th item in T_q.

Example 5: $tu(T_1) = u(a, T_1) + u(c, T_1) + u(d, T_1) = \$14 + \$44 + \$7 = \$65$. The transaction utilities of transactions T_1 to T_{10} are respectively calculated as $tu(T_1) = \$65, tu(T_2) = \$37, tu(T_3) = \$38, tu(T_4) = \$11, tu(T_5) = \$49, tu(T_6) = \$58, tu(T_7) = \$23, tu(T_8) = \$61, tu(T_9) = \$59, and\ tu(T_{10}) = \$42$, as shown in Table VI.

Definition 17: The utility occupancy of an itemset X in a database D_4 is denoted as uo(X), and defined as: the ratio of the utility of X in that transaction divided by the total utility of that transaction:

$$uo(X, T_q) = \frac{u(X, T_q)}{tu(T_q)}$$

Example 6: Since $tu(T_1) = \$65$ and $tu(T_3) = \$38$, the utility occupancy of (ac) in T_1 is calculated as $uo(ac, T_1) = \$58/\$65 \approx 0.8923$, and the utility occupancy of (ac) in T_3 is calculated as $uo(ac, T_3) = \$32/\$38 \approx 0.8421$.

Definition 18: The utility occupancy of an itemset X in a database D_4 is denoted as uo(X), and defined as:

$$uo(X) = \frac{\sum_{X \subseteq T_q \Lambda T_q \in D} uo(X, T_q)}{|\Gamma_X|}$$

where Γ_X is the set of supporting transactions of X in D_4 (thus $|\Gamma_X|$ is equal to the support of X in D_4)

3.5 Occupancy measure in Multi-Objective Task-Oriented Pattern Mining

A multi-objective task-oriented pattern mining problem can be mathematically defined as Maximize $f(x) = (supp(x), \phi(x), area(x))$ where

$$area(X) = \frac{|T_X|.|X|}{\sum_{t \in T_X} |t|}$$

Multi-objective task-oriented pattern mining is to find a set of patterns for maximizing three objectives, namely, support, occupancy, and area.

3.6 Properties of occupancy measure

Property 1: Occupancy is not monotonic or anti-monotonic, convertible and succinct
Property 2: Occupancy is a relative measure
Property 3: Close patterns maximizes occupancy

3.7 Significance of occupancy measure for mining qualified pattern

In the previous section we have seen how occupancy measure is used for different type of databases. An itemset X is a maximal frequent itemset if X is frequent and no superset of X is frequent (Burdick, Calimlim, and Gehrke, 2001). The maximal frequent pattern is the most effective representation since the count of maximal frequent patterns is much smaller when the minimum support is low, which can efficiently reduce the computing cost and storage cost. Furthermore, they are easier to understand for users. A maximal itemset is a largest itemset in a database that is, it is not covered by other itemsets.
We can get multiple maximal frequent itemsets given a support threshold. But we could select the one with the largest number of items as the top qualified pattern. In case of mining maximal frequent patterns, the number of items in a pattern is used as a measure for pattern selection. Therefore, the technique works with the absolute size of the patterns. Unlike maximal frequent patterns occupancy is the relative size of a pattern which is the average ratio of the size of the pattern to the number of items in its supporting sequence. Among all the frequent patterns, the support of maximal frequent pattern selected is usually lower than that of proposed method, thus its precision is lower than the proposed method. Secondly, note that there may exist many different maximal patterns with the same largest length, in which it is difficult to choose one for recommendation by only considering the absolute size of a pattern. In other words, choosing different patterns randomly may occur very different final performance. Lastly, in mining

top qualified pattern a weighted sum of both support and occupancy is used as the interestingness measure, which may lead to better recommendation performance compared to the patterns selected by methods based on maximal frequent patterns.

4. OCCUPANCY BASED PATTERN MINING ALGORITHMS

A concise overview of occupancy based pattern mining algorithms is presented in Figure 1. This figure shows various OPM algorithms, which is compared in the 4.1 subsection.

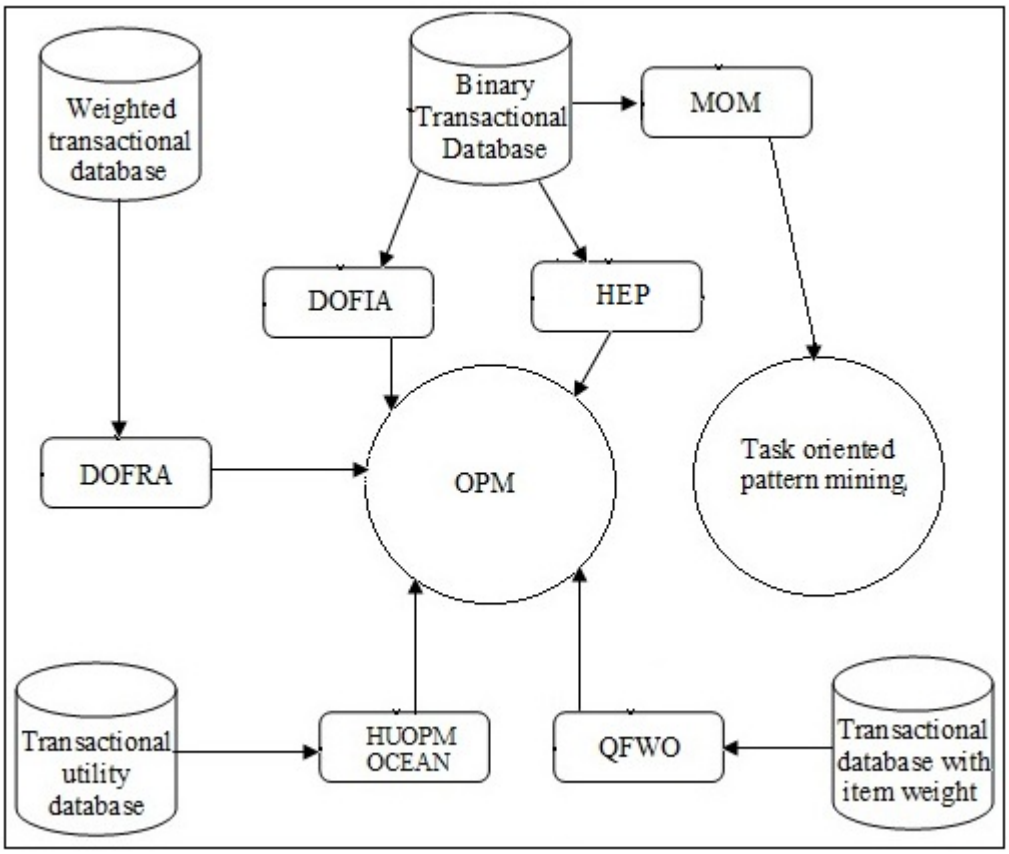

Figure. 1: Framework of occupancy based pattern

4.1 Comparison of state of the art algorithms

A comprehensive review of existing algorithms is presented, with an in-depth discussion of their pros and cons in Table VIII.

Table VIII: Comparison of the Algorithms

Year	**Algorithm**	**Measure**	**Works on/Patterns mined**	**Technique**	**Pros and Cons**
2012	DOFIA (Tang et al., 2012)	Support (α),Occupancy(β), Relative importance of support and occupancy (λ)	Binary transactional database/High quality frequent patterns are mined	Tree based approach	Two pruning techniques (i) Monotonic decreasing property of support value (ii) Occupancy upper bound properties
2015	DOFRA (Zhang et al., 2015)	Support (α), weighted occupancy (β), Relative importance of support and occupancy (λ)	Weighted transactional database/High quality sequential patterns are mined	Prefix-projected database tree	(i)The occupancy and quality of any pattern P in the given subtree will not be larger than its upper bound (ii) A pattern P is closed if P is frequent and there exists no proper super-pattern of P with the same support
2015	HUOPM (Gan et al., 2015)	Support (α), utility occupancy (β), Relative importance of support and occupancy (λ)	Transactional utility database / High utility occupancy patterns are mined	Utility based prefix tree	(i)Two data structures utility occupancy list (UO-list) and frequency-utility table (FU-table) are developed to store the required information about a database to avoid repeated scanning (ii) The algorithm utilizes both the support and utility occupancy measures to prune the search space
2016	OCEAN (Shen et al., 2016)	Support (α), utility occupancy (β), Relative importance of support and occupancy (λ)	Transactional utility database/High utility occupancy patterns are mined	Utility based prefix tree	(i) OCEAN fails to discover the complete set of high utility occupancy patterns
2017	QFWO	Support (α), weight occupancy (β), Relative importance of support and occupancy (λ)	Transactional database with weight of each item/ High weight occupancy patterns are mined	Frequency weight tree	(i)Two compact data structures, weight-list and frequency-weight (FW)-table, are designed to keep essential information about the itemsets from the database. (ii) Algorithm mines patterns from the frequency-weight tree using weight-list with only two database scans
2020	HEP (Deng, 2020)	Support (α), occupancy (β)	Binary transactional database / High occupancy itemsets	Occupancy-list structure	(i)Algorithm generates the occupancy-lists of all 1-itemsets (ii)Then algorithm employs an iterative level-wise approach, where itemsets are used to explore k-itemsets, to discover all high occupancy itemsets

5. WORKING PRINCIPLES OF THE ALGORITHMS

All the algorithms are tree based algorithm.

—DOFIA (DOminant and Frequent Itemset mining Algorithm) adopts lexicographic subset trees to search the frequent itemsets. Frequent pattern mining algorithm leverages the monotonic decreasing property of support value to reduce the search space. That is, if an itemset X is not frequent then all the supersets of X are not frequent either. DOFIA explores the occupancy upper bound properties to drastically reduce the search process, thus can increase the efficiency of mining. The search space is greatly reduced by pruning using the upper bounds for occupancy and quality. Specifically, the algorithm proposed two upper bounds on occupancy for any given itemset X and its supersets in the search tree. The first bound is computationally efficient, while the second is proved to be the tightest bound with certain input constraint. The tradeoff between these two bounds showed in the technique. However, DOFIA can only be applied to transaction databases for itemset mining.

—DOFRA (DOminant and FRequent pattern mining Algorithm) is a tree based Depth-First-Search approach. Each node in the search tree uniquely correspond a pattern. It adopts PrefixSpan algorithm (Pei, Mortazavi, Wang, Pinto, Chen, Dayal, and Hsu, 2004) to find frequent pattern from the sequence database. This algorithm works in different phases. First, a prefix-projected database tree is constructed by the algorithm to find all frequent items. Each node in this tree is a prefix (pattern) projected database which contains three parts: the supporting sequence Sid, the appearance PS (i.e., prefix sequence) of this pattern and the remaining sequence SS (i.e., suffix sequence) behind this pattern in the supporting sequence. After scanning the initial database, all length-1 frequent itemsets are mined. Each pattern can be treated as a prefix and its corresponding projected database is then constructed. Next, these projected databases are used recursively until no frequent items exist in the suffix sequences. When there are no frequent items in supporting suffix sequences, these projected databases cease to expand. The traversal in this database tree is to find all frequent patterns. Two pruning techniques are used by the algorithm to reduce the search space greatly. (i) The occupancy and quality of any pattern P in the given subtree will not be larger than its upper bound. Thus, upper bounds of occupancy and quality are estimated for all frequent patterns in the subtree rooted at P. If the upper bound of occupancy is smaller than the minimal occupancy threshold β, this subtree should be pruned. (ii) A pattern P is closed if P is frequent and there exists no proper super-pattern of P with the same support. All sub-patterns of the closed pattern P with the same support form an equivalence class of P. Among this equivalence class, the closed pattern P must be the one with maximal occupancy value. This property ensures that top qualified pattern must be closed. DOFRA_W is proposed to mine the qualified patterns from weighted sequence database.

—An algorithm called OCEAN was proposed to address the problem of high utility occupancy pattern mining by introducing the utility occupancy measure (Shen, Wen, Zhao, Zhou, and Zheng, 2016). However, it fails to discover the complete set of high utility occupancy patterns (HUOPs) and also encounter several performance problems. First, the mining results derived by OCEAN are incomplete. The reason is that the exact utility information is incorrectly kept in the utility-list structure using an inconsistent sorting order. As a result, OCEAN applies pruning strategies with incorrect information. Second, OCEAN is not efficient as it does not utilize the support property and utility occupancy property well to prune the search space.

—HUOPM (High Utility Occupancy Pattern Mining algorithm) is proposed to mine high utility occupancy patterns with the utility occupancy measure. Two data structures called utility-occupancy list (UO-list) and frequency-utility table (FU-table) are developed to store the required information about a database, for mining HUOP without repeatedly scanning the

database. The remaining utility occupancy is utilized to calculate an upper bound to reduce the search space. HUOPM applies pruning strategies to prune unpromising itemsets early, which greatly speed up the mining efficiency, compared to the OCEAN algorithm. The algorithm utilizes both the support and utility occupancy measures to prune the search space for mining the more interesting and useful high utility occupancy patterns.

—QFWO (highly Qualified pattern with Frequency and Weight Occupancy) is a tree based algorithm to mine highly qualified pattern with a measure called weight occupancy. Two compact data structures, called weight-list and frequency-weight (FW)-table, are designed to keep essential information about the itemsets from the database. The weight list indicates that a list structure which stores some necessary information is associated with an itemset X. Weight-lists allow quick calculation of the weight information of a pattern using join operation. When necessary and sufficient conditions are met, the information of a pattern can be easily obtained from the built weight-lists of its prefix itemsets, thus avoiding repeated database scans. Moreover, the concept of remaining weight occupancy is utilized to calculate the estimated upper bound. Based on the global downward closure property and the partial downward closure property, the QFWO algorithm can directly mine high quality patterns from the frequency-weight tree using weight-list with only two database scans. Without the generation-and-test approach, the QFWO algorithm performs a dept-first search by spanning the FW-tree during construction of the weight-list.

—HEP (High Efficient algorithm for mining high occupancy itemsets) constructs occupancy list (transaction identifier, transaction length) for 1-itemsets. Occupancy list of higher itemsets are constructed without scanning the database by using the intersection of occupancy list of one level lower itemsets. Without generating large numbers of frequent itemsets algorithm employs the upper bound of occupancy as pruning criterion to directly mine high occupancy itemset and avoids costly occupancy computation of numerous frequent itemset.

6. OPEN CHALLENGES AND OPPORTUNITIES

The area of occupancy based pattern mining has been extensively accepted and adopted by the research community recently. The previous section discussed the current status and development of the area of quality pattern mining. However, the existing techniques suffer from certain gaps and drawbacks that need to be resolved for efficiently handling the problem of pattern extraction. This section discusses some challenging issues encountered by the existent occupancy based pattern mining techniques.

—Mining occupancy based patterns from databases with different data characteristics: Real-life databases comprise of data having different data characteristics. The data can be frequent and dense or huge and sparse depending upon the type of application. The sparse databases contain lesser number of these frequently occurring items as compared to the dense databases. It is difficult to get patterns which occupies large portion of the transactions in sparse database. Most occupancy based pattern mining techniques work well with dense datasets having many frequently occurring items but fail to handle the sparse datasets. Therefore, this type of pattern mining techniques must be designed in a way that they can handle both the dense and sparse datasets efficiently. A possible solution could be to use array-based or queue based implementation instead of a tree-based implementation as the performance of tree data structure is not substantial in case of sparse datasets.

—Mining occupancy based patterns from complex data types: So far these types of patterns are extracted from transactions data. Modifications of existing techniques need to be developed for effectively handling the data in other domain such as concept of occupancy can be extended to graph mining (Gleich and Mahoney, 2016), web mining, social media data mining (Barbier

and Liu, 2011) thus is useful in many pattern mining applications.

—Mining occupancy based patterns from large and high dimensional databases: With emerging technology, there is a rapid growth in the size of real world databases. The pattern mining techniques are heavily dependent on main memory and thus become incompatible when it comes to handling large databases. Similarly, working with high dimensional bioinformatics data like microarray and gene expression data is a challenging issue as the datasets contain hundreds and thousands of columns. With increase in row length or the number of columns, the combination of items become exponential, which pose great difficulty in front of the various pattern mining techniques. Thus occupancy based patterns mining techniques are needed that can scale well with increasing database size or dimension.

—Mining occupancy based patterns from dynamic data: The existing occupancy based pattern mining techniques assume the transactional data to be static and operate without considering the dynamic nature of databases. There is a need for expansion of these techniques to work with dynamic databases as well.

7. CONCLUSION

Pattern mining is an active field of research having numerous applications. This survey is an attempt to provide a structured and broad overview of extensive research in the area of occupancy based pattern mining spanning different application domains. In this survey we take an in-depth look at the issues in occupancy based pattern mining, applications of it, current pattern mining approaches, and a discussion on the open challenges in the area. We reviewed the occupancy measure and its applications into different types of databases. We identify the advantages and disadvantages of the various algorithms in each category. The paper discussed the main techniques for exploring the search space of patterns, employed by occupancy based pattern mining algorithms. This article also attempts to provide some significant ideas as feasible future directions for the pattern mining community. Through this brief overview, the article makes an effort to provide some practicable directions to the researchers looking for some viable future perspectives to work on.

References

Adhikari, J. 2014. Mining calendar-based periodic patterns from nonbinary transactions. *Journal of Intelligent Systems Vol.23,* No.3.

Agrawal, R., Imielinski, T., and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Conf. Management of Data.* pp.207–216.

Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan.*

Barbier, G. and Liu, H. 2011. Data mining in social media. In *Social Network Data Analytics.* Springer, 327–352.

Burdick, D., Calimlim, M., and Gehrke, J. 2001. Mafia: A maximal frequent itemset algorithm for transactional databases. In *Proceedings 17th International Conference on Data Engineering.* IEEE.

Cao, Yu, Zhang, and Zhao. 2010. Domain-driven data mining: Challenges and prospects. *IEEE Trans. Knowl. Data Eng. Vol.22,* No.6, 755–769.

Cao, L. 2013. Combined mining: Analyzing object and pattern relations for discovering and constructing complex yet actionable patterns. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov..*

Deng, Z. 2020. Mining high occupancy itemsets. *Future Gener. Comput. Syst. Vol.102*, 222–229.

Gade, K., Wang, J., and Karypis, G. 2004. Efficient closed pattern mining in the presence of tough block constraints. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, USA, August*. ACM, 138–147.

Gan, W., Lin, J. C., F.Viger, P., Chao, H., Tseng, V. S., and Yu, P. S. 2018. A survey of utility-oriented pattern mining. *CoRR abs/1805.10511*.

Gan, W., Lin, J. C., F.Viger, P., Chao, H., Zhan, J., and Zhang, J. 2018. Exploiting highly qualified pattern with frequency and weight occupancy. *Knowl. Inf. Syst. Vol.56,* No.1, 165–196.

Gan, W., Lin, J. C. W., F.Viger, P., Chao, H. C., and Yu, P. S. 2020. HUOPM: high utility occupancy pattern mining. *IEEE Trans. Cybern. Vol.50,* No.3, 1195–1208.

Gleich, D. F. and Mahoney, M. W. 2016. Mining large graphs. In *Handbook of Big Data*. Chapman and Hall/CRC, 191–220.

Haff, L. R. 1979. An identity for the wishart distribution with applications. *Journal of Multivariate Analysis*.

Moens, S., Aksehirli, E., and Goethals, B. 2013. Frequent itemset mining for big data. In *BigData*. pp.111–118.

M.S.Chen, Han, J., and P.S.Yu. 1996. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering Vol.8,* No.6.

Pei, J., Mortazavi, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. 2004. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. Knowl. Data Eng.*.

P.F.Viger, P. Lin, J. C., Vo, B.and Chi, T. T. Z., and Le, H. 2017. A survey of itemset mining. *WIREs: Data Mining and Knowledge Discovery Vol.7,* No.4.

Shen, B., Wen, Z., Zhao, Y., Zhou, D., and Zheng, W. 2016. Ocean: Fast discovery of high utility occupancy itemsets. In *Advances in Knowledge Discovery and Data Mining, PAKDD*. Springer, 354–365.

Tang, L., Zhang, L., Luo, P., and Wang, M. 2012. Incorporating occupancy into frequent pattern mining for high quality pattern recommendation. In *Proc. of the 21st ACM International Conference on Information and Knowledge Management*. ACM, pp.75–84.

Wang, L., Meng, J., Xu, P., and Peng, K. 2018. Mining temporal association rules with frequent itemsets tree. *Applied Soft Computing Vol.62,* No.

Yao, H., Hamilton, H. J., and Butz, C. J. 2004. A foundational approach to mining itemset utilities from databases. 482–486.

Zhang, X., Duan, F., Zhang, L., Cheng, F., Jin, Y., and Tang, K. 2017. Pattern recommendation in task-oriented applications: A multi-objective perspective [application notes]. *IEEE Comput. Intell. Mag.*.

Dr. Jhimli Adhikari received Master of Computer Application and Ph D in Computer Science from Jadavpur University, Kolkata and Goa University, respectively. At present, she is Associate Professor in the Department of Computer Science, Narayan Zantye College, Goa, India. She has twenty two years of teaching experience. Her areas of research interest include data mining and knowledge discovery, decision support systems and data science. She is coauthor of two research monographs and published six international journal papers, three international conference papers, one book chapter and one book review. She is regular reviewer of Pattern Recognition Letters, Elsevier and Editorial board member of International Journal of Image Processing and Pattern Recognition, JournalsPub.