A Method of Mining Spatial High Utility Co-location Patterns Based on Feature Actual Participation Weight

Xin Zeng, Jian Yang, Zhenpeng Li and Xiaowei Li*

College of Mathematics and Computer, Dali University, Dali, Yunnan, 671003, China

*Corresponding author’s e-mail: lixiaowei_xidian@163.com

Abstract. Spatial high utility co-location patterns mining takes pattern utility participation index as interest measure, but it still lacks standard method for calculating pattern utility participation index. This paper presents some reasonable definitions of spatial high utility co-location patterns based on the datasets of instance with utility value, such as Feature Actual Participation Weight, Feature Utility Participation Index and Pattern Utility Participation Index and many more. A basic algorithm BWHMA(high utility co-location patterns mining based on feature actual participation weight) is proposed. For improving the efficiency of algorithm BWHMA, one pruning algorithm FUPRA(based on feature utility participation rate) and one optimization algorithm FRPWA(based on feature actual participation weight) are put forward. Finally, a large number of comparative experiments have been done on both synthetic and real datasets. The experiments show that the proposed algorithms are feasible, correct and valid.

1. Introduction

Research on mining frequent patterns from traditional transactional databases is relatively mature. However with the rapid development of information network technology and the emergence of a large number of mobile terminals, spatial data show explosive growth and spatial databases are also widely used. Due to characteristics of spatial data, such as massiveness, high dimensionality and complexity etc. [1], traditional pattern mining algorithms are difficult to apply to spatial data mining directly. How to discover potential, interesting spatial relationships or valuable patterns from spatial databases and guide users to make scientific decisions has become one of the research hotspots in the field of data mining.

Spatial co-location pattern is a subset of spatial features. Spatial co-location pattern mining uses pattern participation index as interest measure, without considering the profit (also known as the utility value) of the feature instance. Spatial high utility co-location patterns mining uses pattern utility as interest, which introduces pattern utility participation index as mining condition and gets the co-location patterns with high degree of feature utility participation.

There are some metrics of spatial high utility pattern, Yang et al. used the unit profit of feature $f_i$ as external utility $u(f_i)$ [2]. The number of distinct instances of feature $f_i$ in the table instances of pattern $c$ as internal utility $q_i$, so feature utility is $v(f_i) = u(f_i) \times q_i$. All instances of $f_i$ have the same external utility, which are shown in Table 1.
### Table 1 Unit profit of four crops

| $f_i$ | A | B | C | D |
|-------|---|---|---|---|
| unit profit | 8 | 4 | 1 | 30 |

### Table 2 Row instance and participation index

| $A$ | $B$ | $A$ | $C$ | $B$ | $C$ | $A$ | $B$ | $C$ |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1   | 1   | 1   | 1   | 1   | 1   | 11  | 1   | 11  |
| 4   | 2   | 3   | 3   | 2   | 2   | 42  | 2   | 2   |
| 4   | 2   | 23  | 23  | 23  | 23  | 12  | 12  | 12  |
| 1/2 | 3   | 1   | 11  | 1   | 11  | 123 | 1   | 1   |
| 1/2 | 3/4 | 1   | 1   | 11  | 11  |

Spatial neighbor relationships of four crops are shown in figure 1.

![Figure 1 Spatial neighbor relationships of four crops](image)

![Figure 2 Spatial instances with utility value](image)

In order to judge $c = \{A, B\}$ is a high utility co-location pattern. Firstly, we should calculate the table instance of $c$ in figure 1, $table\_instance(c) = \{[A1,B1],[A4,B2]\}$. Secondly, we should calculate pattern utility $u(c) = 8 \times 2 + 4 \times 2 = 24$ and database utility $U(S) = 8 \times 4 + 4 \times 2 + 1 \times 3 + 30 \times 2 = 103$. Finally, the utility rate of $c$ is $\lambda(c) = u(c) / U(S) = 0.23$. If the user sets the pattern utility rate threshold $\xi = 0.2 \leq \lambda(c) = 0.23$, $c$ is a spatial high utility co-location pattern.

Ref. [3] takes the instance with utility value as the research object. The utility participation rate $UPR(f_i, c)$ of $f_i \in c$ is defined as the weighted sum of the internal utility rate ($IntraUR$) and the inter utility rate ($InterUR$) of the feature $f_i$, the complete calculation formula is $UPR(f_i, c) = w_1 \times IntraUR(f_i, c) + w_2 \times InterUR(f_i, c)$. $w_1$ and $w_2$ represent the weight of $IntraUR$ and $InterUR$ of feature $f_i$, respectively. $w_1$ and $w_2$ are given by users. But all features have the same weight, which is not in accordance with the actual situation. In order to solve the problem of weight, Wang et al. [4] proposed spatial high utility co-location pattern mining method based on feature utility participation rate. Firstly, the utility participation rate $FUR(f_i, c)$ of $f_i$ equals that the utility sum of the distinct instances of $f_i$ in $table\_instance(c)$ divides the utility sum of all instances of $f_i$.

Secondly, we should calculate the utility weight $w(f_i, c)$ of $f_i$, i.e., $w(f_i, c) = \frac{FUR(f_i, c)}{\sum_{f_j \in c} FUR(f_j, c)}$.

Finally, the pattern utility participation index $PUI(c)$ equals the sum of the utility participation rate multiplies the utility weight of each feature, i.e., $PUI(c) = \sum_{f_i \in c} w(f_i, c) \times FUR(f_i, c)$. If $PUI(c) \geq \xi$, $c$ is a high utility co-location pattern.

This paper proposes a method of mining spatial high utility co-location patterns based on feature actual participation weight. The method will further improve the feature utility participation weight.
proposed in Ref.[2, 3] and give the actual weight of $f_i$. The main contributions of this paper are as follows:

1. A more reasonable calculation method of the actual participation index of $f_i$ in pattern $c$ based on the feature utility participation rate is proposed and the metric method of high utility pattern is given.
2. A basic algorithm is proposed, then we make a pruning algorithm based on the basic algorithm and we also give an optimization algorithm of mining high utility co-location pattern and prove the correctness of algorithm.
3. Numerous experiments have been done on scalability of algorithm based on “real + synthetic” datasets.

The remainder of this paper is organized as follows: Section 2 presents the related research work of data mining. The basic knowledge of co-location pattern mining is presented in Section 3. In Section 4, we give the definition and characteristic of high utility co-location pattern mining. Section 5 presents the basic algorithm, pruning algorithm and optimization algorithm of high utility co-location pattern mining. The experiments and analysis are given in Section 6. Finally, the conclusion and future work are given in Section 7.

2. Related work

Apriori [5] and FP-Growth [6] have been widely used in the field of data mining. The mining research works have been achieved fruitful results on determining datasets, fuzzy datasets, and spatial datasets. Huang et al. [7] proposed the Join-Based algorithm of the Apriori-like algorithm and interest measure, namely the concept of participation rate and participation index. In order to reduce the connection between instances, the Partial-based algorithm and Joinless algorithm were proposed in Ref.[8, 9]. The traditional framework of mining prevalent co-location patterns produces numerous redundant co-location patterns. To address this issue, Ref.[10] studied the problem of reducing redundancy in a collection of prevalent co-location patterns by utilizing the spatial distribution information of co-location instances. Segregation patterns have not received much attention. A method for finding both types of interaction based on a statistical test was proposed in Ref.[11].

In recent years, high utility co-location pattern mining has become a research hotspot. High utility itemsets, which can be exploited, lack inherent structural properties. Ref.[12] aimed to improve the state-of-the-art and proposes a high utility mining method that employs novel pruning strategies. Ref.[13] proposed a new framework for top-k high utility itemsets mining to address the issues about a user-specified minimum utility threshold. High utility pattern mining suffers from the scalability issue due to the huge number of candidates. A high utility itemset growth approach has been proposed without generating candidates by Ref.[14]. In real-world application, transactions are changed whether insertion or deletion in a dynamic database. An existing maintenance approach for handling high-utility itemsets in dynamic databases with transaction deletion must rescans the database when necessary. An efficient algorithm, called PRE-HUI-DEI, for updating high-utility itemsets based on the pre-large concept for transaction deletion was proposed in Ref.[15]. Ref.[16] proposed that the new data points are continuously increased in dynamic database. It is not necessary to scan the entire updated datasets, only the table instance of the changed part is calculated.

3. Co-location pattern related concepts and properties

Four spatial features $A, B, C, D$ have 4, 2, 3, 2 instances, respectively. The spatial distribution and neighbor relationships of all instances are presented in figure 1. Each instance consists of an instance number, a feature type, and a position coordinate, e.g., $A.2$ indicates that the instance of feature $A$ at a certain location is 2. The instance $A.2$ and $D.2$ are connected by solid lines to indicate that they satisfy the spatial neighbor relationship $R : R(A.2, D.2) \iff distance(A.2, D.2) \leq d$, $d$ is a pre-set distance threshold. The distance between the instances is calculated using the Euclidean distance. The instance set is $I = \{i_1, i_2, \cdots, i_n\}$. If any two instances in the instance set $I$ satisfies
The spatial co-location pattern represents a subset of the spatial feature set, denoted by \( c \). The number of feature in \( c \) are its length, e.g., \( \{A,B,C\} \) is a pattern of length 3. A longer pattern containing \( c \), called a superset of \( c \), e.g., \( \{A,B,C\} \) is superset of \( \{A,B\} \). The number of feature in \( c \) is \( 1 \), and any subsets of them does not contain all the objects. Each row instance is a clique, denoted by \( \text{row_instance}(c) \). A collection of all row instances of \( c \) called a table instance, denoted by \( \text{table_instance}(c) \). The row instance and table instance of some patterns are listed in table 2.

Suppose \( f_i \) is a spatial feature. The participation rate \( PR(c,f_i) \) of \( f_i \) in \( c \) is expressed as:

\[
PR(c,f_i) = \pi_i \left( \text{table_instance}(c) \right) / \left| \text{table_instance}(f_i) \right|
\]

\( \pi \) is the projection operation. \( \text{table_instance}(f_i) \) denotes the table instance of \( f_i \). \( \pi_i \left( \text{table_instance}(c) \right) \) denotes the number of the distinct instance of \( f_i \) in the table instance of \( c \). The participation index of \( c \) is the minimum of the participation rates of all spatial features, i.e.,

\[
\Pi(c) = \min_{f_i \in c} \left\{ PR(c,f_i) \right\}
\]

e.g., \( \min_{prev} = 0.6 \), \( \Pi(\{A,C\}) = 0.75 \) and \( \Pi(\{B,C\}) = 1 \) are all greater than \( \min_{prev} \) in table 2, so \( \{A,C\} \) and \( \{B,C\} \) are frequent patterns. \( \Pi(\{A,B\}) = 0.5 \) and \( \Pi(\{A,B,C\}) = 0.5 \) are less than \( \min_{prev} \), so \( \{A,B\} \) and \( \{A,B,C\} \) are not frequent patterns.

**Theorem 1**
The participation index and participation rate monotonically decrease as the length of \( c \) increases.

Proof: Suppose \( c = \{f_1, f_2, \ldots, f_k\} \), it is easy to proof that. The proof process is as follows:

\[
\Pi(c \cup f_i) = \min_{\sum_{j=1}^{k} f_{i+1}} \left\{ PR(c \cup f_{i+1}, f_i) \right\} \leq \min_{\sum_{j=1}^{k} f_{i+1}} \left\{ PR(c \cup f_{i+1}, f_i) \right\} \leq \min \left\{ PR(c, f_i) \right\} = \Pi(c)
\]

4. Related definition and properties of high utility co-location pattern

4.1 Related definition of high utility pattern

**Definition 1 Spatial instance with utility value** [3] The \( j \)-th instance with utility value \( v \) of \( f_i \) is denoted as \( f_i^{v_j} \) or \( u(f_i, j) = v \), e.g., the first instance \( D.1 \) of \( D \) has a utility value 6, i.e., \( D.1^6 \).

**Definition 2 Feature utility** If \( f_i \) contains \( k \) instances, feature utility \( FU(f_i) \) is the sum of utility value of all instances, i.e., \( FU(f_i) = \sum_{j=1}^{k} u(f_i, j) \). The utility value of each instance and all feature utility are presented in table 3.

| Feature | Utility value | Feature utility |
|---------|---------------|----------------|
| A       | 3, 5, 7, 11   | 26             |
| B       | 10, 18        | 28             |
| C       | 27, 33, 30    | 90             |
| D       | 6, 10         | 16             |

**Definition 3 Feature utility participation rate** The sum of utility value of the distinct instance of \( f_i \) in \( \text{table_instance}(c) \) divides \( FU(f_i) \), i.e., \( FUPR(f_i, c) = \frac{\sum_{j=1}^{k} u(f_i, j)}{FU(f_i)} \),
\( f_i, j \in \pi_f \), \( \text{table\_ins\ tan\ ce\( c(c) \)} \). The table instance of \( c = \{A, B\} \) is \( \left[ \left[ \{A,1,1,1,1\}, \{A,4,1,1,1\} \right] \right] \) in figure 2, so \( \text{FUPR}(A, c) = \frac{3+11}{26} = 0.54 \) and \( \text{FUPR}(B, c) = \frac{10+18}{28} = 1 \).

**Definition 4 Pattern actual utility** If the length of \( c \) is \( k \), the sum of utility value of the distinct instance of \( f_i \) in \( \text{table\_ins\ tan\ ce\( c(c) \)} \) is pattern actual utility \( \text{PRU}_c \), i.e., \( \text{PRU}_c = \sum_{i=1}^{k} \sum u(f_i, j) \), \( f_i, j \in \pi_f \), \( \text{table\_ins\ tan\ ce\( c(c) \)} \). If \( c = \{B, C\} \) in figure 2, \( \text{PRU}_c = 10 + 27 + 18 + 33 = 88 \).

**Definition 5 Feature actual participation weight** The sum of utility value of the distinct instance of feature \( f_i \) in \( \text{table\_ins\ tan\ ce\( c(c) \)} \) is divided by \( \text{PRU}_c \), i.e., \( \text{FRPW}_c = \sum_{i=1}^{k} \sum u(f_i, j) \), \( f_i, j \in \pi_f \), \( \text{table\_ins\ tan\ ce\( c(c) \)} \). For example, if \( c = \{B, C\} \) in figure 2, \( \text{FRPW}_{B, c} = \frac{10+18}{88} = 0.32 \) and \( \text{FRPW}_{C, c} = \frac{27+33}{88} = 0.68 \). According to the definitions of pattern actual utility and feature actual participation weight, we know that the sum of feature actual participation weight of all the feature in \( c \) equals to 1, i.e., \( \sum_{i=1}^{k} \text{FRPW}(f_i, c) = 1 \).

**Proof:**

\[
\sum_{i=1}^{k} \text{FRPW}(f_i, c) = \sum_{i=1}^{k} \frac{\sum u(f_i, j)}{\text{PRU}_c} = \frac{\sum_{i=1}^{k} \sum u(f_i, j)}{\text{PRU}_c} = \frac{\sum_{i=1}^{k} \sum u(f_i, j)}{\sum_{i=1}^{k} u(f_i, j)} = 1 , \quad f_i, j \in \pi_f \text{table\_ins\ tan\ ce\( c(c) \)} .
\]

For example, if \( c = \{B, C\} \), the sum of all feature in \( c \) equals to 1, i.e., \( \text{FRPW}(B, c) + \text{FRPW}(C, c) = 1 \).

**Definition 6 Feature utility participation index** The feature utility participation index \( \text{FUPI}_c \) of \( f_i \) in \( c \) equals \( \text{FUPR}(f_i, c) \) multiplied by \( \text{FRPW}(f_i, c) \).

**Definition 7 Pattern utility participation index** The pattern utility participation index \( \text{PUPI}_c \) is minimum among feature utility participation index of all features in \( c \), i.e., \( \text{PUPI}_c = \min \left( \text{FUPI}(f_i, c) \right), 1 \leq i \leq k \). For example, in figure 2, the feature utility participation rate and feature actual participation weight in \( c_1 = \{A, B\} \) are \( \text{FUPR}(A, c_1) = 0.54 \), \( \text{FUPR}(B, c_1) = 1 \), \( \text{FRPW}(A, c_1) = 0.33 \), \( \text{FRPW}(B, c_1) = 0.67 \), respectively. Finally, \( \text{PUPI}_1 = \min (0.54 \times 0.33, 1 \times 0.67) = 0.18 \). If \( c_2 = \{B, C\} \), then \( \text{FUPR}(B, c_2) = 1 \), \( \text{FUPR}(C, c_2) = 0.67 \), \( \text{FRPW}(B, c_2) = 0.32 \) and \( \text{FRPW}(C, c_2) = 0.68 \), respectively. Finally, \( \text{PUPI}_2 = \min (0.32, 0.67) = 0.32 \). Suppose the user-set pattern utility participation index threshold is \( \text{pupi\_th} = 0.3 \). Since \( \text{PUPI}_1 \) is less than \( \text{pupi\_th} \), \( c_1 \) is not a high utility co-location pattern. \( \text{PUPI}_2 \) is greater than \( \text{pupi\_th} \), \( c_2 \) is a high utility co-location pattern.

### 4.2 Related properties

**Theorem 2** Pattern utility participation index does not satisfy anti-monotonicity: the pattern utility participation index \( \text{PUPI}_c \) does not monotonically decrease as the length of pattern increases.
**Proof:** Suppose $c' \subset c$, then the number of the row instance meets the following condition: 
\[ \text{count}(\text{row_ins}(c')) \geq \text{count}(\text{row_ins}(c)). \]
According to the clique of row instance, each row instance of $c$ must contain a row instance of $c'$. If $f_i \in c \cap c'$, $\text{FUPR}(f_i, c')$ is greater than $\text{FUPR}(f_i, c)$. But the pattern utility participation index is related to feature utility participation rate and feature actual participation weight. Since the utility value of the instances of $f_j \in c$ and $f_j \notin c'$ are uncertain, $\exists u(f_j, x)$ satisfies $\text{PUPI}(c) \geq \text{PUPI}(c')$.

5. **Algorithm**

In this section, we will present and analyze the pruning and optimization algorithm base on the basic algorithm.

5.1 **Basic algorithm**

**Algorithm 1 BWHMA**

| Input: Feature set $F = \{f_1, f_2, \ldots, f_n\}$, set of instances with utility value $IU$, distance threshold $d$, pattern utility participation index threshold $\text{pupi}_\text{th}$. |
|---|
| Output: High utility co-location pattern set $HUP$. |
| Variable: $FU$ : Feature utility; $c_i$ : order candidate pattern set; $c$ : Single pattern in $c_i$; $\text{FUPR}$ : Feature utility participation rate; $\text{PRU}$ : Pattern actual utility; $\text{FRPW}$ : Feature actual participation weight; $\text{FUPI}$ : Feature utility participation index; $\text{PUPI}$ : Pattern utility participation index. |

1. $HUP = \emptyset$;
2. $FU(f_i) = \text{comp}_\text{FU}(F, IU)$;
3. $C_2 = \text{gen}_\text{cand}_\text{patt}(F)$;
4. For $k = 2$; $C_k \neq \emptyset$; $k + +$  
5. do
6. repeat
7. $c \in C_k$;
8. $\text{tab}(c) = \text{gen}_\text{tab}_\text{ins}(c, IU, d)$;
9. For each $f_i \in c$
10. do
11. $\text{FUPR}(f_i, c) = \text{comp}_\text{FUPR}(\text{tab}(c), FU(f_i))$;
12. $\text{PRU}(c) = \text{comp}_\text{PRU}(\text{tab}(c))$;
13. $\text{FRPW}(f_i, c) = \text{comp}_\text{FRPW}(\text{tab}(c), \text{PRU}(c))$;
14. $\text{FUPI}(f_i, c) = \text{FUPR}(f_i, c) \times \text{FRPW}(f_i, c)$;
15. done
16. $\text{PUPI}(c) = \text{min}(\text{FUPI}(f_i)), 1 \leq i \leq k$;
17. if $\text{PUPI}(c) \geq \text{pupi}_\text{th}$ then
18. $HUP = HUP \cup c$;
19. until all $c$ has been traversed in $C_k$;
20. $C_{k+1} = \text{gen}_\text{cand}_\text{patt}(C_k)$;
21. done

step 2) Calculating the feature utility; step 3) Producing candidate pattern set of length 2; step 4) If $C_k$ is not empty and $c \in C_k$, calculating $\text{table}_\text{ins} \text{tan} \text{ce}(c)$ step 8); Calculating the feature utility participation rate, feature utility, feature actual participation weight, and feature utility participation index of each feature in steps 10-15); step 16) The minimum feature utility participation index is taken as pattern utility participation index; step 17) If the pattern utility participation index of $c$ is greater than or equal to the pattern utility participation index threshold given by users, $c$ is incorporated into high utility pattern set $HUP$ in step 18); When $C_k$ is traversed in step 19), $C_{k+1}$ is generated based on $C_k$ in step 20). Return to step 4) continues to loop until $C_k$ is empty.

5.2 **Pruning algorithm based on feature utility participation rate**
Theorem 3 Suppose $c = \{f_i, \cdots, f_k\}$, if $\exists f_i \in c$ and $FUPR(f_i, c) \leq \text{pupi\_th}$, $c$ is not a high utility co-location pattern and its superset is also not a high utility co-location pattern.

Proof: Suppose $c = \{f_i, f_j, \cdots, f_k\}, k \geq 2$, and pattern utility participation index threshold is $\text{pupi\_th}$.

According to the definition of feature actual participation weight, we know $FRPW(f_i, c) < 1$. If $\exists f_i \in c$ and $FUPR(f_i, c) \leq \text{pupi\_th}$, then $FUPR(f_i, c) \times FRPW(f_i, c) < \text{pupi\_th}$. $PUPI(c)$ is the minimum of feature utility participation index. If the feature utility participation rate of any feature in $c$ is less than or equals to pattern utility participation index threshold, $c$ is not a high utility co-location pattern. If $c' = c \cup f_{k+1}$ and $c \subset c'$, then $\text{count} \left( \text{row\_ins}(c') \right) \leq \text{count} \left( \text{row\_ins}(c) \right)$. If $f_i \in c \cap c'$, $FUPR(f_i, c') \leq FUPR(f_i, c) \leq \text{pupi\_th}$ is established. When $FUPR(f_i, c') \leq \text{pupi\_th}$, $c'$ is not a high utility co-location pattern. $c'$ can be directly pruned. Next, the pruning algorithm FUPRA is obtained based on the basic algorithm BWHMA, which is directly updated in steps 9-16 and other steps are unchanged.

Algorithm 2 FUPRA

10-1. for each $f_i \in c$ 16-1. else
11-1. do 17-1. $\text{PRU}(c) = \text{comp\_PRU}(\text{tab}(c))$;
12-1. $\text{FUPR}(f_i, c) = \text{comp\_FU}$ 18-1. $\text{FRPW}(f_i, c) = \text{comp\_FRPW}(\text{tab}(c), \text{PRU}(c))$;
13-1. $\left( \text{tab}(c), \text{FU}(f_i) \right)$; 19-1. $\text{FUPR}(f_i, c) = \text{FUPR}(f_i, c) \times \text{FRPW}(f_i, c)$;
14-1. if $\text{FUPR}(f_i, c) \leq \text{pupi\_th}$ then 20-1. done
15-1. $C_i = C_i - c$; 21-1. $\text{PUPI}(c) = \text{min}(\text{FUPI}(f_i)), 1 \leq i \leq k$.
18-1. goto step 19; //Jump to step 19 of the basic algorithm
21-1. $\text{PUPI}(c) = \text{min}(\text{FUPI}(f_i)), 1 \leq i \leq k$.

Pruning $c$ with a feature utility participation rate less than or equals to $\text{pupi\_th}$ in step 14-1) of FUPRA. The number of candidate pattern of length $k+1$ generated by the connection in step20) of BWHMA is reduced. The number of pattern in $C_{k+1}$ is less, which improves the execution efficiency of FUPRA.

5.3 Optimization algorithm based on feature actual participation weight

Firstly, the pattern, which the feature utility participation rate is less than or equals to the pattern utility participation index threshold, is pruned based on FUPRA. Secondly, we use optimization algorithm FRPWA based on the feature actual participation weight to prune for the remaining patterns.

Lemma 1 If $\exists f_i \in c$ and $\text{FRPW}(f_i, c) < \text{pupi\_th}$, $c$ is not a high utility co-location pattern.

Proof: After the pruning algorithm FUPRA is completed, the feature utility participation rate of any feature of $c$ in the candidate pattern set $C_i$ satisfies $\text{pupi\_th} < \text{FUPR}(f_i, c) \leq 1$.

If $\text{FRPW}(f_i, c) < \text{pupi\_th}$, then $\text{FUPI}(f_i, c) = \text{FUPR}(f_i, c) \times \text{FRPW}(f_i, c) < \text{pupi\_th}$. But $\text{FUPI}(c) = \text{min}(\text{FUPI}(f_i)), 1 \leq i \leq k$, so $\text{FUPI}(c) < \text{pupi\_th}$ and $c$ is not a high utility co-location pattern.

The input, output and variables of the optimization algorithm FRPWA are consistent with the basic algorithm BWHMA. The optimization algorithm FRPWA is obtained only on the pruning algorithm FUPRA (step 10-1 to 21-1).

Algorithm 3 FRPWA

10-2. for each $f_i \in c$ 16-2. $\text{PRU}(c) = \text{comp\_PRU}(\text{tab}(c))$;
11-2. do
In step 18-2), if \( \exists \text{FRPW}(f_i,c) < \text{pupi}_\text{th} \), then \( c \) is not a high utility pattern. The execution step will jump directly to judge the next pattern. The optimization algorithm FRPWA reduces the number of calculations and further improves the execution efficiency.

6. Experiment and analysis

In this section, we will compare the scalability of algorithm BWHMA and FPRWA on synthetic datasets and real datasets. For the different calculation methods of pattern utility participation index, we give the comparative analysis of FPRWA and MaxFURAS in ref.[4]. The hardware and software environment is as follows: Intel Core i7 CPU, 8GB RAM, 250G SSD; Windows 10 operating system, Python 3.6 programming language.

6.1 Scalability of algorithm under synthetic datasets

Synthetic datasets used in this paper has 15 features. The number, coordinates and utility value of the instance are randomly generated. The coordinate range and utility value range of the instance are \( 1000 \times 1000, [1, 1000] \).

6.1.1 Scalability of algorithm by datasets size

Synthetic datasets size is 156, 405, 600, 809, and 1018, respectively. When the distance threshold is 30 and the pattern utility participation index threshold is 0.06, we test the scalability of algorithm by datasets size. The experiments are shown in figure 3.

![Figure 3 Scalability of FPRWA by datasets size](image)

In figure 3, the number of the instance continues to increase to result in a growing datasets. The calculation of the table instance of pattern increases, too. This means that the execution time of BWHMA and FPRWA also increases. However, FPRWA prunes some patterns and reduces the computational amount of the table instance. Therefore, FPRWA is more efficient than BWHMA.

6.1.2 Scalability of algorithm by distance threshold

Datasets size is 600. The pattern utility participation index threshold is 0.06. When the distance threshold is 10, 20, 30, 40, and 50, we test the scalability of BWHMA and FPRWA, respectively.

![Figure 4 Scalability of FPRWA by distance threshold](image)
The experiments are shown in figure 4. As the distance threshold increases, more pairs of instance become neighbor pairs. The number of table instance increases, and the time consumption in experiments increases. The FPRWA algorithm prunes some patterns and their supersets based on the feature utility participation rate, which reduces the instance connection and makes the operation more efficient.

6.1.3 Scalability of algorithm by pattern utility participation index threshold

Datasets size is 600. The distance threshold is 30. When the pattern utility participation index threshold is 0.02, 0.04, 0.06, 0.08, and 0.10, we test the execution efficiency of BWHMA and FPRWA, respectively. The experiments are shown in figure 5.

![Figure 5 Scalability of FPRWA by pattern utility participation index threshold](image1)

![Figure 6 Scalability of FPRWA by datasets size on real datasets](image2)

When the pattern utility participation index threshold is continuously increased, the feature participation rate of more patterns is less than or equal to the pattern utility participation index threshold. These patterns will be pruned by FPRWA algorithm. The execution time continues to decrease. For BWHMA, since the number of patterns that need to calculate the table instance is not decreased, the execution time is almost unchanged.

6.2 Scalability of algorithm under real datasets

The real datasets is obtained from Chongqing POI data, which includes 15 characteristics such as gourmet stores, hotels, leisure and entertainment places, companies, and shopping stores, et al. The number of instance and the range of utility value are shown in Table 4.

| Name of datasets | Number of instance | Range of utility value |
|------------------|--------------------|------------------------|
| real153          | 153                | (1, 100000)            |
| real401          | 401                | (1, 100000)            |
| real603          | 603                | (1, 100000)            |
| real801          | 801                | (1, 100000)            |
| real1003         | 1003               | (1, 100000)            |

Firstly, for testing the scalability of BWHMA and FPRWA by real datasets size, distance threshold and pattern utility participation index are set to 0.02 and 0.08, respectively. The experiments are shown in figure 6. Secondly, for testing the scalability of BWHMA and FPRWA by distance threshold, we select real603 as test datasets and pattern utility participation index threshold is set to 0.08. The experiments are shown in figure 7. Finally, for test the scalability of BWHMA and FPRWA by pattern participation index threshold, real603 is used as a test datasets and distance threshold is set to 0.02. The experiments are shown in figure 8.
6.3 Comparative analysis of FPRWA and MaxFURAS

Although FPRWA and MaxFURAS all adopt the concept of feature utility participation weight in the calculation method of pattern utility participation index, the calculation method of feature utility participation weight is completely different. Next, we select real603 as test dataset. Distance threshold is set to 0.02. When the pattern utility participation index threshold is 0.02, 0.04, 0.06, 0.08, and 0.10, we compare and analysis the execution efficiency and the number of high utility co-location pattern of FPRWA and MaxFURAS. The experiments are shown in figure 9 and figure 10.

Both FPRWA and MaxFURAS use feature utility participation rate to reduce branches. If any of feature utility participation rates is less than or equal to the pattern utility participation index threshold, the pattern will be pruned by FPRWA. If all of feature utility participation rate are less than or equal to the pattern utility participation index threshold, the pattern will be pruned by MaxFURAS. So for the same pattern utility participation index threshold, FPRWA has a larger pruning range and higher execution efficiency than MaxFURAS.

The pattern utility participation index in FPRWA is the minimum value of the feature utility participation index, but in MaxFURAS it is the sum of the utility participation index of all features. Since our comparison based on the fact that applying the same pattern utility participation index threshold on the two algorithms, the results suggest that MaxFURAS mines more high utility co-location patterns than FPRWA.

7. Conclusion and Future work

In this paper, we present a new definition of spatial high utility co-location patterns based on the instances with utility value, and set the new pattern utility participation index metrics. Based on the new definition and metrics, BWHMA, FUPRA and PPRWA are proposed. Extensive experimental results show that the proposed algorithms outperform other algorithms found in the literature. An important contribution of our approach is that it drastically reduces the CPU time associated with other
algorithms. We demonstrate the effectiveness of our algorithm using both synthetic and real datasets. Future work includes applying these algorithms to dynamic databases and fuzzy datasets, to conform the experimental results in the real life domain. Furthermore, proposing an efficient way of turning the algorithm’s parameters which would result in better solutions, and modifying our algorithms to effectively and efficiently applied to large-scale distributed data mining will be our future working directions.

Acknowledgement
This research was partially supported by the National Natural Science Foundation of China (NSFC) under grant No.61462003, No.71462001, No.71661001, and by the Application Foundation Youth Project of Yunnan Provincial Science and Technology Department under grant No.2016FD071.

References
[1] Han, J., Kamber, M. (2006) Data mining: concepts and techniques (Third Edition). China Machine Press, Beijing.
[2] Yang, S.S., Wang, L.Z., Lu, J.L., Gao, Y. (2014) Primary exploration for mining spatial high utility co-location patterns. Journal of Chinese Mini-Micro Computer Systems, 35: 2302-2307.
[3] Jiang, W.G., Wang, L.Z., Fang, Y., Chen, H.M. (2017) Domain-driven high utility co-location pattern mining method. Journal of Computer Applications, 37: 322-328.
[4] Wang, X.X., Wang, L.Z., Chen, H.M, Fang, Y., Yang, P.Z. (2018) Mining spatial high utility co-location patterns based on feature utility ratio. Chinese Journal of Computers, 41: 1-18.
[5] Agrawal, R., Imielinski, T., Swami, A. (1993) Mining association rules between sets of items in large databases. In: ACM SIGMOD Conference. Washington. pp. 207-216.
[6] Han, J., Pei, J., Yin, Y. (2000) Mining frequent patterns without candidate generation. In: ACM SIGMOD Conference. Texas. pp. 1-12.
[7] Huang, Y., Shekhar, S., Xiong, H. (2004) Discovering co-location patterns from spatial datasets: A general approach. IEEE Transactions on Knowledge and Data Engineering, 16: 1472-1485.
[8] Yoo, J.S., Shekhar, S., Smith, J., Kumquat, J.P. (2004) A partial join approach for mining co-location patterns. In: Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems. New York. pp. 241-249.
[9] Jin, S.Y., Shekhar, S. (2006) A joinless approach for mining spatial co-location patterns. IEEE Transaction on Knowledge and Data Engineering, 18: 1323-1337.
[10] Wang, L.Z., Bao, X.G., Zhou, L.H. (2018) Redundancy reduction for prevalent co-location patterns. IEEE Transactions on Knowledge and Data Engineering, 30: 142-155.
[11] Barua, S., Sander, J. (2014) Mining statistically significant co-location and segregation patterns. IEEE Transactions on Knowledge and Data Engineering, 26: 1185-1199.
[12] Krishnamoorthy, S. (2015) Pruning strategies for mining high utility itemsets. Expert Systems with Applications, 42: 2371-2381.
[13] Aiman, A., Gulmeher, R. (2018) Efficient algorithms for mining Top-K High Utility Itemsets. International Journal of Computer Sciences and Engineering, 6: 1274-1280.
[14] Liu, J., Wang, K., Fung, B.C.M. (2012) Direct discovery of high utility itemsets without candidate generation. In: IEEE 12th International Conference on Data Mining. Brussels. pp. 984-989.
[15] Lin, C.W., Hong, T.P., Lan, G.C., Wong, J.W., Lin, W.Y. (2015) Efficient updating of discovered high-utility itemsets for transaction deletion in dynamic databases☆. Advanced Engineering Informatics, 29: 16-27.
[16] Wang, X.X., Wang, L.Z., Lu, J.L., Zhou, L.H. Effectively Updating High Utility Co-location Patterns in Evolving Spatial Database. International Conference on Web-age Information Management. Nanchang. pp. 67-81.