

Article

# Normalized-Mutual-Information-Based Mining Method for Cascading Patterns

Cunjin Xue <sup>1,2,\*</sup>, Jingyi Liu <sup>1</sup>, Xiaohong Li <sup>1</sup> and Qing Dong <sup>1,\*</sup>

<sup>1</sup> Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China; jingyiliu24@163.com (J.L.); li\_xiaohong2011@163.com (X.L.)

<sup>2</sup> Key Laboratory of the Earth Observation, Sanya 572029, Hainan, China

\* Correspondence: xuecj@radi.ac.cn (C.X.); qdong@radi.ac.cn (Q.D.); Tel.: +86-10-8217-8126 (C.X)

Academic Editors: Ozgun Akcay and Wolfgang Kainz

Received: 24 May 2016; Accepted: 19 September 2016; Published: 27 September 2016

**Abstract:** A cascading pattern is a sequential pattern characterized by an item following another item in order. Recent research has investigated a challenge of dealing with cascading patterns, namely, the exponential time dependence of database scanning with respect to the number of items involved. We propose a normalized-mutual-information-based mining method for cascading patterns (M<sup>3</sup>Cap) to address this challenge. M<sup>3</sup>Cap embeds mutual information to reduce database-scanning time. First, M<sup>3</sup>Cap calculates the asymmetrical mutual information between items with one database scan and extracts pair-wise related items according to a user-specified information threshold. Second, a one-level cascading pattern is generated by scanning the database once for each pair-wise related item at the quantitative level. Third, a recursive linking–pruning–generating loop generates an  $(m + 1)$ -level-candidate cascading pattern from  $m$ -dimensional patterns on the basis of antimonotonicity and non-additivity, repeating this step until no further candidate cascading patterns are generated. Fourth, meaningful cascading patterns are generated according to user-specified minimum evaluation indicators. Finally, experiments with remote sensing image datasets covering the Pacific Ocean demonstrate that the computation time of recursive linking and pruning is significantly less than that of database scanning; thus, M<sup>3</sup>Cap improves performance by reducing database scanning while increasing intensive computing.

**Keywords:** spatiotemporal data mining; cascading pattern; mutual information; marine remote sensing

## 1. Introduction

A cascading pattern means that the occurrence and development of one parameter always has an impact on the second parameter, followed by the third, and so on [1]. In geoinformatics, such patterns are common. For example, in ocean science, an El Niño Southern Oscillation (ENSO) might be followed by the abnormal changes of sea surface temperature (SST), followed by a sea level anomaly (SLA), and finally by sea surface precipitation (SSP) [2]. This phenomenon can be represented in the form of “ENSO → SST → SLA → SSP”, where “→” means followed by. Similarly, as an example involving natural hazards, typhoons are always followed by rainstorms and then by floods, in the form of “Typhoon → Rainstorm → Flood” [3]. Such a followed by reaction is a specific type of sequential pattern, which was first introduced by Agrawal and Srikant [4].

In recent decades, several methods have been developed to capture sequential patterns, which can be organized into two categories. The first category focuses on designing methods and improving their performance, which includes Apriori-based and pattern-growth algorithms [5]. The second category

focuses on the specified applications by revising prevalent algorithms (e.g., mobile patterns [6–8], geographic events [9,10], and cascading patterns [11]).

Apriori was the first algorithm proposed by Agrawal and Srikant to deal with market basket data analysis [4]. The generalized sequential pattern (GSP) algorithm was then developed as an improvement [12]. Such algorithms use the core idea of Apriori and adopt the recursive linking and pruning function, which requires the repetition of database scanning to generate sequential patterns. To overcome this problem using predefined data formats, new data types have been introduced to replace items in generating sequential patterns (e.g., SPADE with event ID [13] and SPaMi-FTS with bit-vectors [14]). In essence, Apriori-based algorithms use breadth-first search and Apriori pruning, which generate large sets of candidates for growing longer sequences [15]. To reduce the large sets of candidates, pattern-growth algorithms have been developed with the core idea of divide and conquer [16–19]. As the complete set of sequential patterns can be partitioned into different subsets according to different prefixes, the number of candidates decreases substantially but at the cost of increasing the corresponding projected databases. With several frequent sequences, both categories of algorithm fail to run efficiently because a larger number of candidate subsequences require a larger number of database scans or projected database constructions.

For real applications, some typical algorithms have been revised to meet specified requirements. For example, by defining a spatial sequence index, Huang et al. [9] proposed a temporal-slicing-based algorithm to find sequential spatiotemporal events, which involve Apriori-based linking and pruning. The graph-based mining algorithm adopted the core idea of divide and conquer by replacing items with trajectory information lists to generate all frequent trajectory patterns by linking and pruning [6]. QPrefixSpan—a direct extension of PrefixSpan—was proposed to discover the frequent sequential patterns of quantitative temporal events [10]. For cascading patterns, the cascading spatiotemporal pattern miner (CSTPM) adopted filtering strategies to reduce the number of repetitions of database scanning, but the core idea was breadth-first search, followed by Apriori pruning [11]. Generally, these methods resolve real applications in several domains, and with little ability to improve mining algorithm performance.

Faced with a large amount of repetitions of database scanning to generate meaningful frequent patterns and an intensive computing to generate candidates for frequent patterns, both two categories of methods have great challenges to discover our proposed cascading patterns. Our one-level cascading pattern in the form of “ $X_1 \rightarrow X_2$ ” has a meaning similar to that of a pair-wise related item obtained from normalized mutual information. Normalized mutual information tells the amount of information one item provides about the other [20]. As an asymmetry, normalized mutual information can produce the causal effect relationship between items, which has been widely used to find associated or correlated patterns in data mining [21–24]. However, in real applications, the cascading pattern, in a form of “ $X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_n$ ”, is quite different from the associated pattern, in a form of “ $X_1 X_2 \dots X_k \rightarrow X_{k+1} X_{k+2} \dots X_n$ ”. For example, “La Niña  $\rightarrow$  SST  $\rightarrow$  SLA” means that a La Niña event will induce SST occurrence, and then followed by SLA, while “La Niña  $\rightarrow$  SST, SLA” means that a La Niña event will induce a co-occurrence of SST and SLA.

Thus, the motivation behind this paper is to design a method for mining cascading patterns effectively by embedding normalized mutual information, thereby reducing the number of database scans required and improving performance. Our study aims for the following.

1. We propose an algorithm—normalized-mutual-information-based mining method for cascading patterns (M<sup>3</sup>Cap)—for mining cascading patterns by embedding normalized mutual information. M<sup>3</sup>Cap uses asymmetric mutual information from items to obtain pair-wise related items initially, which substantially reduces the number of database scans required to generate cascading patterns.
2. We analyze the computational complexity of M<sup>3</sup>Cap and classify it into two categories—database scanning and intensive computing. The former category obtains meaningful cascading patterns through evaluation indicators and the latter category obtains all candidates for a cascading pattern using recursive linking and pruning.

- Real datasets with long-term remote sensing images covering the Pacific Ocean are used to evaluate the performance of M<sup>3</sup>Cap. Mutual information, support, and cascading index thresholds primarily affect the repetitions of database scanning, and the number of database records affects the time for one database scan. The number of derived items and mutual information threshold determine the intensive computing.

The remainder of this paper is organized as follows. In Section 2, we discuss some of the concepts and properties that relate to the use of mutual information in the cascading pattern-mining algorithm. In Section 3, we describe the M<sup>3</sup>Cap workflow and address four key issues. In Section 4, experiments with remote sensing image datasets are described to discuss M<sup>3</sup>Cap's computational complexity, and the main factors affecting its performance are analyzed. We discuss and summarize our study in Section 5.

## 2. Basic Concepts and Properties

This study uses the following terminology.

An item is a type of geographical environmental parameter or phenomenon (e.g., temperature, precipitation, primary productivity, normalized difference vegetation index, or an ENSO event).

Normalized mutual information is the amount of information one item ( $X$ ) provides about another ( $Y$ ) [21], as shown in Equations (1)–(3).

$$I(X; Y) = \frac{\sum_{v_x \in \text{dom}(x)} \sum_{v_y \in \text{dom}(y)} p(v_x, v_y) \times \log \left[ \frac{p(v_x, v_y)}{p(v_x) \times p(v_y)} \right]}{\sum_{v_x \in \text{dom}(x)} p(v_x) \times \log [p(v_x)]} \quad (1)$$

$$p(v_x) = \frac{n(v_x)}{N} \quad (2)$$

$$p(v_x, v_y) = \frac{n(v_x, v_y)}{N} \quad (3)$$

where  $X$  and  $Y$  are the items,  $N$  is the total number of transaction records, and  $v_x$  and  $v_y$  are quantitative levels in domain  $\text{dom}(x)$  and  $\text{dom}(y)$ , respectively.  $n(v_x)$  and  $n(v_x, v_y)$  are the occurrence number of  $v_x$  and the co-occurrence number of  $v_x$  and  $v_y$ , respectively.  $p(v_x)$  is the probability density of item  $X$ , and  $p(v_x, v_y)$  is the joint probability density between items  $X$  and  $Y$ .

Support and confidence are the common evaluation indicators of association rules [25]. In this paper, we use those concepts to define two new evaluation indicators for cascading patterns—support and cascading index.

Support is the co-occurrence probability of items  $X_1, X_2, \dots, X_m$ , in the dataset, which is defined as

$$S(X_1 X_2 \dots X_m) = \frac{n(X_1 X_2 \dots X_m)}{N} \times 100\% \quad (4)$$

where  $N$  is the total number of the dataset, and the  $n(X_1, X_2, \dots, X_m)$  is the co-occurrence number of items  $X_1, X_2, \dots, X_m$ , in the dataset.

Cascading index describes the occurrence probability of item  $X_m$  assuming that item  $X_{m-1}, X_{m-2}, \dots, X_1$  occur in order, defined as:

$$CI(X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_m) = \min \left\{ \frac{n(X_1, X_2, \dots, X_{k+1})}{n(X_1, X_2, \dots, X_k)} \times 100\%, \quad 1 \leq k \leq m-1 \right\} \quad (5)$$

where  $k$  is a cascading level and  $m$  is the number of items.

A cascading pattern is an ordered relationship with sequential items shown in the following form, in which  $S$  and  $CI$  are evaluation indicators.

$$X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_m (S\%, CI\%) \quad (6)$$

An  $m$ -level cascading pattern is a cascading pattern consisting of  $(m + 1)$  items.

$(m - 1)$ -level subsets of an  $m$ -level cascading pattern are the subsets of an  $m$ -level cascading pattern at level  $(m - 1)$ . There are  $(m + 1)$  cascading patterns of  $(m - 1)$ -level subsets of an  $m$ -level cascading pattern.

**Example 1.** When La Niña events occur in the western Pacific Ocean, reinforced by the North and South Equatorial Currents and the Equatorial Counter Current, the sea surface temperature (SST) rises abnormally, the increasing SST accumulates water mass moving westward, and SLA rises abnormally. Thus, the cascading pattern among La Niña, SST, and SLA is in the form of “La Niña  $\rightarrow$  SST  $\rightarrow$  SLA.” This pattern is a two-level cascading pattern with three one-level subpatterns, “La Niña  $\rightarrow$  SST”, “La Niña  $\rightarrow$  SLA”, and “SST  $\rightarrow$  SLA”.

**Example 2.** Supposed that the two-level cascading pattern “La Niña  $\rightarrow$  SST  $\rightarrow$  SLA” in Example 1 is meaningful, and the records in database are shown in Table 1, where 1 means that a La Niña event occurs, SST rises, or SLA rises, while 0 means it does not, and ID is a uniform identifier that represents a particular record—a regular field in a database table. Thus, the Support of “La Niña  $\rightarrow$  SST  $\rightarrow$  SLA” is  $n(\text{La Niña, SST, SLA})/N \times 100\%$  (i.e., 30%), and the Cascading Index is the minimum  $\{n(\text{La Niña, SST})/n(\text{La Niña}) \times 100\%, 66.7\%, n(\text{La Niña, SST, SLA})/n(\text{La Niña, SST}) \times 100\%, 75.0\%, \text{ (i.e., 66.7\%)}\}$ .

As normalized mutual information is asymmetrical, related items have a directionality of the form “ $X_1 \rightarrow X_2$ ”, and the normalized mutual information matrix of the items identifies only one-level cascading patterns. Obtaining the full set of cascading patterns requires linking and pruning, which depend on the properties of the normalized mutual information.

**Property 1. Information Asymmetry.** Asymmetry is the property by virtue of which the information that item  $X_1$  provides to item  $X_2$  differs from the information provided by item  $X_2$  to item  $X_1$ . This property provides the pair-wise related items with directionality (e.g., “ $X_1 \rightarrow X_2$ ” may be true in particular conditions, meaning that a variation in  $X_1$  will promote changes in  $X_2$ , while “ $X_2 \rightarrow X_1$ ” may not be true). This property has been demonstrated by Ref. [21].

**Property 2. Information Non-Additivity.** Non-additivity is the property by virtue of which when dealing with multiple-level processes, information decreases after each level. That is to say, information processing simply transforms the signal, data, or message into a useful form, creating no new information. This property is shown as the following inequalities.

$$\left. \begin{array}{l} I(X_1; X_2; \cdots; X_m) \leq I(X_2; X_3; \cdots; X_m) \\ I(X_1; X_2; \cdots; X_m) \leq I(X_1; X_3; \cdots; X_m) \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ I(X_1; X_2; \cdots; X_m) \leq I(X_1; X_2; \cdots; X_{m-1}) \end{array} \right\} \quad (7)$$

**Proof of Property 2.** Given a two-level cascading pattern of the form “ $X_1 \rightarrow X_2 \rightarrow X_3$ ,” denoted as  $I(X_1; X_2; X_3)$ , meaning that  $X_1$  indicates the amount of information when  $X_3$  occurs after  $X_2$  occurs. According to the related concepts and definitions of Reference [20], the following Equations (8) and (9) are true.

$$I(X_1; X_2; X_3) = I(X_1; X_2 X_3) - I(X_1; X_3 | X_2) \quad (8)$$

$$I(X_1; X_2) = I(X_1; X_2 X_3) - I(X_1; X_2 | X_3) \quad (9)$$

where,  $I(X_1; X_2 X_3)$  represents a joint mutual information among  $X_1$ ,  $X_2$ , and  $X_3$ , meaning that  $X_1$  tells about the amount of information when  $X_2$  and  $X_3$  co-occur, while  $I(X_1; X_3 | X_2)$  represents a conditional mutual information, meaning that given  $X_2$ , the amount of information  $X_1$  tells about  $X_3$ . As given  $X_2$ ,  $X_1$ , and  $X_3$  are independent, that is,  $I(X_1; X_2 | X_3) = 0$ , Equation (10) is true.

$$I(X_1; X_2; X_3) = I(X_1; X_2) - I(X_1; X_3 | X_2) \quad (10)$$

As  $I(X_1; X_3 | X_2) \geq 0$ , Formula(11) is true.

$$I(X_1; X_2; X_3) \leq I(X_1; X_2) \quad (11)$$

Formulas (12) and (13) also follow from similar reasoning,

$$I(X_1; X_2; X_3) \leq I(X_1; X_3) \quad (12)$$

$$I(X_1; X_2; X_3) \leq I(X_2; X_3) \quad (13)$$

**Property 3. Antimonotonicity.** All nonempty subsets of a cascading pattern must also be cascading patterns.

**Proof of Property 3.** The support and cascading index of a cascading pattern of the form “ $X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_m$ ” satisfy the Equation (14). Considering its  $(m - 1)$  level subpatterns, according to Equations (5) and (6), the Equations (15) and (16) are true. Thus, all the  $(m - 1)$  level subsets are cascading patterns. The  $(m - 2)$  to one-level subpatterns can also be proved using a similar workflow.

$$\left. \begin{array}{l} S(X_1 X_2 \cdots X_m) \geq \min \text{Support} \\ CI(X_1 X_2 \cdots X_m) \geq \min \text{CascadingIndex} \end{array} \right\} \quad (14)$$

$$\left. \begin{array}{l} S(X_2 X_3 \cdots X_m) \geq S(X_1 X_2 \cdots X_m) \\ S(X_1 X_3 \cdots X_m) \geq S(X_1 X_2 \cdots X_m) \\ \dots \dots \dots \dots \dots \dots \dots \dots \\ S(X_1 X_2 \cdots X_{m-1}) \geq S(X_1 X_2 \cdots X_m) \end{array} \right\} \quad (15)$$

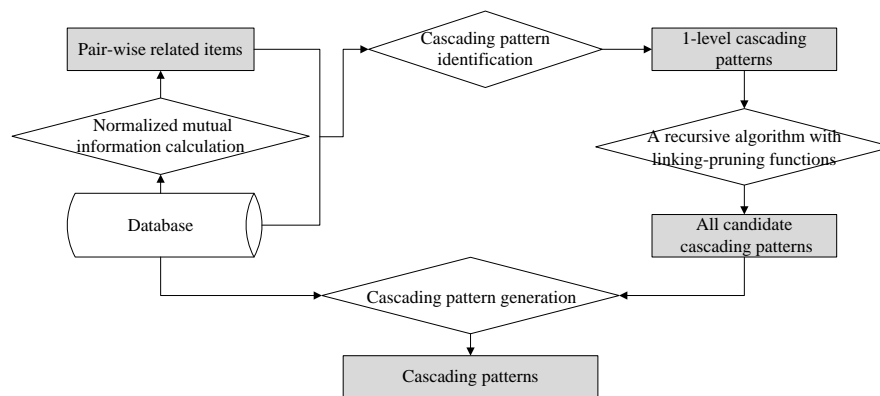
$$\left. \begin{array}{l} CI(X_2 X_3 \cdots X_m) \geq CI(X_1 X_2 \cdots X_m) \\ CI(X_1 X_3 \cdots X_m) \geq CI(X_1 X_2 \cdots X_m) \\ \dots \dots \dots \dots \dots \dots \dots \dots \\ CI(X_1 X_2 \cdots X_{m-1}) \geq CI(X_1 X_2 \cdots X_m) \end{array} \right\} \quad (16)$$

**Table 1.** Boolean dataset for Example 2.

ID Item	0	1	2	3	4	5	6	7	8	9
La Niña	1	1	1	1	0	0	0	1	1	0
SST	1	1	0	1	0	0	1	0	1	0
SLA	1	1	1	1	0	0	1	0	0	0

### 3. Normalized-Mutual-Information-Based Mining Method for Cascading Patterns (M<sup>3</sup>Cap)

Using the basic concepts and properties described above, the M<sup>3</sup>Cap workflow, which involves four key steps, is shown in Figure 1. The first step calculates the normalized mutual information among items and yields pair-wise related items. The second step generates one-level cascading patterns from pair-wise related items. The third designs recursive linking-pruning functions to generate all candidate cascading patterns. The fourth step retains the meaningful cascading patterns by removing the pseudo patterns from candidates.



**Figure 1.** Normalized-mutual-information-based mining method for cascading patterns Workflow.

### 3.1. An Algorithm for Producing Pair-Wise Related Items

On the basis of discretization datasets (i.e., a mining transaction table), the database was scanned once to calculate the normalized mutual information between all pairs of items according to Equations (1)–(3), and their normalized mutual information contingency table was constructed. Table 2 shows its structure. Each element in this table,  $I(X_i, X_j)$ , indicates that item  $X_i$  provides information about item  $X_j$ , which is represented in the form “ $X_i \rightarrow X_j$ ”.

**Table 2.** Structure of normalized mutual information contingency table.

	$X_1$	$X_2$	$X_3$	...	$X_M$
$X_1$	1.000	$I(X_1, X_2)$	$I(X_1, X_3)$	...	$I(X_1, X_M)$
$X_2$	$I(X_2, X_1)$	1.000	$I(X_2, X_3)$	...	$I(X_2, X_M)$
$X_3$	$I(X_3, X_1)$	$I(X_3, X_2)$	1.000	...	$I(X_3, X_M)$
...	...	...	...	...	...
$X_m$	$I(X_M, X_1)$	$I(X_M, X_2)$	$I(X_M, X_3)$	...	1.000

In general, the mean and standard deviation of the items’ mutual information are used to reveal highly related items [23]. Here, only the elements within the table fulfilling the following inequality are regarded as pair-wise related items.

$$I(X_i; X_j) \geq \mu + \lambda \sigma \quad (17)$$

where,  $\mu$  and  $\sigma$  are the mean and standard deviation of the mutual information when  $i$  is not equal to  $j$ , respectively, and  $\lambda$  is a scalar factor. Owing to the asymmetry of the pair-wise related item, “ $X_i \rightarrow X_j$ ”, we define  $X_i$  as an antecedent and  $X_j$  as a consequent.

### 3.2. One-Level Cascading Patterns with Quantitative Ranks

This step produces all the one-level cascading patterns with quantitative ranks from pair-wise related items. Quantitative ranks represent an item’s variations. In general, not all of the pair-wise related items with quantitative ranks are meaningful. That is to say, only the item’s quantitative ranks with a frequency not less than the evaluation thresholds are retained. Thus, this issue requires three steps. The first step extracts meaningful related items with quantitative ranks by scanning the database once. The second step generates candidate one-level quantitative cascading patterns through the permutation and combination of the meaningful items with quantitative ranks. The third step obtains one-level cascading patterns from candidates by scanning the database using the following criterion, i.e., a frequency not less than the user-specified support and a cascading index not less than the user-specified threshold. The database is scanned once for each candidate one-level quantitative cascading pattern. Thus, all one-level cascading patterns are obtained. Algorithm 1 shows the pseudo-code for this function.



---

**Algorithm 1.** Algorithm for identifying one-level cascading patterns from pair-wise related items.

---

**Algorithm name:** OneLevelCascadingPatternIdentification.

**Algorithm description:** Identifying all one-level cascading patterns with quantitative ranks from pair-wise related items.

**Input parameters:** Pair-wise related items, denoted as  $PI$ , and a quantitative rank, denoted as  $R$ .

**Output parameters:** One-level cascading patterns, denoted as 1-CP.

---

```

1  OneLevelCascadingPatternIdentification ( $PI$ ,  $R$ , 1-CP)
2  FOR  $i$ th pair-wise related item in  $PI$ , denoted as  $i$ th –  $pi$ .
3      Extract its antecedent and consequent, denoted as  $X_i$  and  $X_j$ , respectively
4
5      Permute and combine the  $X_i$  and  $X_j$  with quantitative ranks, i.e.,  $r$  and  $l$ , respectively,
6      where  $r, l \in R$ , then generate candidates of cascading pattern in a form of  $X_i[r] \rightarrow X_j[l]$ 
7      FOR each  $X_i[r] \rightarrow X_j[l]$ 
8          Scan the database one time, and calculate the supports of  $X_i$  and  $X_j$  with
9          quantitative ranks through the Equation (4).
10         IF the following inequalities are true, where  $\tau_S$  is the support threshold.
11
12             
$$\left. \begin{array}{l} S(X_i[r]) \geq \tau_S \\ S(X_j[l]) \geq \tau_S \end{array} \right\}$$

13             The  $X_i$  and  $X_j$  with the quantitative rank  $r$  and  $l$  are retained, i.e.,  $X_i[r]$ 
14             and  $X_j[l]$ .
15         END IF
16         Using Equations (4) and (5), calculate its support and cascading index.
17
18         IF the following inequalities are true, where  $\tau_{CI}$  is the cascading index threshold.
19
20             
$$\left. \begin{array}{l} S(X_i[r]X_j[l]) \geq \tau_S \\ CI(X_i[r]X_j[l]) \geq \tau_{CI} \end{array} \right\}$$

21             Link  $X_i[r]$  and  $X_j[l]$  in a form with  $X_i[r] \rightarrow X_j[l]$ , denote it as one-level
22             cascading pattern, i.e., 1- $cp$ , and store it into 1-CP.
23
24         END IF
25     END FOR
26      $i = i + 1$ 
27 END FOR
```

---

Line 10 to Line 12 are discriminant criteria, which are used to find the meaningful items with a specified quantitative rank (Line 13 to Line 14), similarly, Line 18 to Line 20 are done to generate a new one-level cascading pattern. From Line 7 to Line 25, a pair-wise related item with all quantitative ranks is gone through to generate zero to several one-level cascading patterns, and Line 2 to Line 27 are repeated to deal with all pair-wise related items.

**Example 3.** Given pair-wise related items  $X$  and  $Y$ ,  $X$  and  $Z$ , and  $Y$  and  $Z$  of the forms “ $X \rightarrow Y$ ”, “ $X \rightarrow Z$ ”, and “ $Y \rightarrow Z$ ” are meaningful, and their quantitative ranks set to 5 from  $-2$  to  $2$  with a continuous interval. Quantitative datasets are shown in Table 3. The support and cascading index thresholds are set to 30% and 60%, respectively.

According to Table 3, the frequencies of  $X[2]$ ,  $X[1]$ ,  $X[0]$ ,  $X[-1]$ , and  $X[-2]$  are 50%, 0%, 10%, 30%, and 10%, respectively, and the frequencies of  $Y[2]$ ,  $Y[1]$ ,  $Y[0]$ ,  $Y[-1]$ , and  $Y[-2]$  are 40%, 30%, 20%, 0%, and 10%, respectively, while those of  $Z[2]$ ,  $Z[1]$ ,  $Z[0]$ ,  $Z[-1]$ , and  $Z[-2]$  are 0%, 0%, 20%, 30%, and 50%, respectively. According to the support threshold, only  $X[2]$ ,  $X[-1]$ ,  $Y[2]$ ,  $Y[1]$ ,  $Z[-1]$ , and  $Z[-2]$  are meaningful. With the pair-wise related items, “ $X \rightarrow Y$ ,” and through their permutation and combination, the candidate one-level cascading patterns are “ $X[2] \rightarrow Y[2]$ ”, “ $X[2] \rightarrow Y[1]$ ”, “ $X[-1] \rightarrow Y[2]$ ”, and “ $X[-1] \rightarrow Y[1]$ ”. After applying the user-specified support and cascading index thresholds, only the one-level cascading pattern, “ $X[2] \rightarrow Y[2]$ ,” which has a support of 30% and a cascading index of 60%, is retained. “ $X[2] \rightarrow Z[-2]$  (40%,80%)” and “ $Y[2] \rightarrow Z[-2]$  (40%,100%)” are obtained similarly.

**Table 3.** Quantitative dataset in database for Example 3.

Item \ ID	ID									
	0	1	2	3	4	5	6	7	8	9
X	2	2	2	2	−1	−1	−1	2	0	−2
Y	2	2	0	2	1	0	2	1	1	−2
Z	−2	−2	−2	−2	−1	−1	−2	0	0	−1

### 3.3. Recursive Algorithm for Generating the Candidates of All Cascading Patterns

This recursive algorithm produces all the candidate  $(m + 1)$ -level cascading patterns from the  $m$ -level patterns according to the linking and pruning functions, where  $m$  is not less than one. In this step, the linking and pruning functions are run recursively without scanning the database until no further candidate cascading patterns are generated.

The linking function generates the candidate  $(m + 1)$ -level cascading patterns from the one-level and  $m$ -level patterns. The pseudo-code for the linking function is shown in Algorithm 2.

---

**Algorithm 2.** Algorithm of linking  $m$ -level cascading patterns to generate  $(m + 1)$ -level ones.

---

**Algorithm name:** Linking Algorithm.

**Algorithm description:** Link  $m$ -level cascading patterns (i.e.,  $m$ -CP) to generate  $(m + 1)$ -level cascading patterns (i.e.,  $(m + 1)$ -CP) according to the Property 1 Information Asymmetry.

**Input parameters:** One-level cascading patterns, 1-CP, and  $m$ -level cascading patterns,  $m$ -CP.

**Output parameters:**  $(m + 1)$ -level cascading patterns,  $(m + 1)$ -CP.

---

```

1  LinkingAlgorithm (1-CP,  $m$ -CP,  $(m + 1)$  - CP)
2  FOR  $i$ th  $m$ -level cascading pattern in  $m$ -CP, denoted as  $i$ th  $m$ -cp.
3      Extract its last item, i.e.,  $X_{m+1}^i[r_{m+1}^i]$ , and denote it as PostItem, and the others are
4      organized in order in a form with  $X_1^i[r_1^i] X_2^i[r_2^i] \dots X_m^i[r_m^i]$ , denoted as PreItems.
5      FOR  $j$ th 1-level cascading pattern in 1-CP, denoted as  $i$ th 1-cp.
6          Extract its antecedent, i.e.,  $X_1^j[r_1^j]$ , denoted as Pre, and consequent,  $X_2^j[r_2^j]$ ,
7          denoted as Post.
8          IF Pre is same to PostItem, i.e., the items  $X_1^j$  and  $X_{m+1}^i$  are same, and the levels
9           $r_1^j$  and  $r_{m+1}^i$  are equal. AND not find  $X_2^j$  in  $X_1^i[r_1^i] X_2^i[r_2^i] \dots X_m^i[r_m^i]$ .
10             Link  $m$ -cp with appending the item  $X_2^j[r_2^j]$  in a form with  $X_1^i[r_1^i] \rightarrow X_2^j[r_2^j]$ 
11              $X_2^j[r_2^j] \rightarrow \dots \rightarrow X_m^i[r_m^i] \rightarrow X_{m+1}^i[r_{m+1}^i] \rightarrow X_2^j[r_2^j]$ , denote it as  $(m + 1)$ -cp,
12             and store it into  $(m + 1)$ -CP
13         END IF
14      $j = j + 1$ 
15 END FOR
16  $I = I + 1$ 
17 END FOR

```

---



Lines 8 and 9 provide the linking discrimination criteria. Linkage is performed if and only if the last item of the  $m$ -level cascading pattern is the same as the first item of a one-level cascading pattern (i.e., they have the same single item and the same quantitative level). The  $m$ -level cascading pattern does not contain the last item of the one-level cascading pattern (Line 10 to Line 12). From Line 8 to Line 13, zero (the discrimination criterion is false) to one (the discrimination criterion is true) linkage is performed. From Line 5 to Line 15, a loop between one  $m$ -level cascading pattern with all of the one-level patterns is executed, and Line 2 to Line 17 are repeated to generate all of the  $(m + 1)$ -level cascading patterns. The key issue in this function is the linking discrimination.

**Example 4.** There are one-level cascading patterns “ $X[2] \rightarrow Y[2]$ ”, “ $X[2] \rightarrow Z[-2]$ ”, and “ $Y[2] \rightarrow Z[-2]$ ” (Example 3 results). As the cascading patterns are one-level, the input  $m$ -CP is also one-CP (i.e., they have the same cascading patterns). For the first cascading pattern in  $m$ -CP, “ $X[2] \rightarrow Y[2]$ ”, the function goes through the one-CP sequentially. As the first cascading pattern in one-CP is the same as “ $X[2] \rightarrow Y[2]$ ”, no linkage is performed. Regarding the second pattern in one-CP, “ $X[2] \rightarrow Z[-2]$ ”, as the last item  $Y[2]$  is not the same as the first item  $X[2]$ , no linkage is performed. Regarding the third in one-CP, “ $Y[2] \rightarrow Z[-2]$ ”, as the last item  $Y[2]$  is the same as the first item  $Y[2]$ , one linkage is performed and a new form is generated, “ $X[2] \rightarrow Y[2] \rightarrow Z[-2]$ ”. Similarly, the other cascading patterns in  $m$ -CP are analyzed, and there is no linkage. Thus, there is only one two-level cascading pattern generated from the one-level cascading patterns, “ $X[2] \rightarrow Y[2] \rightarrow Z[-2]$ ”.

The pruning function removes the false  $m$ -level cascading patterns from the candidate  $m$ -level patterns according to Properties 2 and 3. The key issue of this step is to find all  $(m - 1)$ -level subpatterns of cascading patterns of an  $m$ -level pattern. As long as one subpattern is not an  $(m - 1)$ -level cascading pattern, the  $m$ -level candidate cascading pattern is removed. Otherwise, the candidate cascading pattern is retained. The pseudo-code for the pruning function is shown as Algorithm 3.

---

**Algorithm 3.** Pruning algorithm to remove the pseudo cascading patterns.

---

**Algorithm name:** PruningAlgorithm.

**Algorithm description:** Remove the pseudo cascading patterns from the  $m$ -level candidates of cascading patterns, denoted as  $m$ -CCP, to generate  $m$ -level cascading patterns, denoted as  $m$ -CP, according to the Property 2. Information Non-Additivities and Property 3. Antimonotonicity.

Input parameters:  $m$ -level candidates of cascading patterns (i.e.,  $m$ -CCP) and  $(m - 1)$ -level cascading patterns, denoted as  $(m - 1)$ -CP.

Output parameters:  $m$ -level cascading patterns (i.e.,  $m$ -CP).

---

```

1  PruningAlgorithm ( $m$ -CCP,  $(m - 1)$ -CP,  $m$ -CP)
2  FOR  $i$ th  $m$ -level candidate of cascading pattern in  $m$ -CCP, denoted as  $i$ th  $m$ -ccp.
3      Extract its  $(m - 1)$ -level subsets, denoted as  $(m - 1)$ -S.
4      FOR  $j$ th  $(m - 1)$ -level cascading pattern in  $(m - 1)$ -S, denoted as  $j$ th  $(m - 1)$ -s
5          IF  $j$ th  $(m - 1)$ -s is not found in  $(m - 1)$ -CP
6              GO TO Line 11
7          END IF
8           $J = j + 1$ 
9      END FOR
10   $i$ th  $m$ -ccp is retained as a  $m$ -level cascading pattern, and stored into  $m$ -CP.
11   $i$ th  $m$ -ccp is removed
12   $I = I + 1$ 
13  END FOR

```

---

Line 5 is the removal discrimination criterion. As long as one  $(m - 1)$ -level subpattern of the candidate  $m$ -level pattern is not found among the  $(m - 1)$ -level cascading patterns, the candidate  $m$ -level cascading pattern is removed (Line 11) and the next candidate  $m$ -level cascading pattern

is processed (Line 12). If all the  $(m - 1)$ -level subsets of the candidate  $m$ -level cascading pattern are  $(m - 1)$ -level cascading patterns, the candidate  $m$ -level cascading pattern is taken as  $m$ -level one (Line 10). From Line 4 to Line 9, a loop is executed through all of the  $(m - 1)$ -level subsets sequentially to determine the cascading patterns. Line 3 extracts all of the  $(m - 1)$ -level subpatterns of an  $m$ -level cascading pattern, and Line 2 to Line 13 are repeated to distinguish all of the candidate  $m$ -level cascading patterns.

### 3.4. Generating Cascading Patterns with Evaluation Indicators

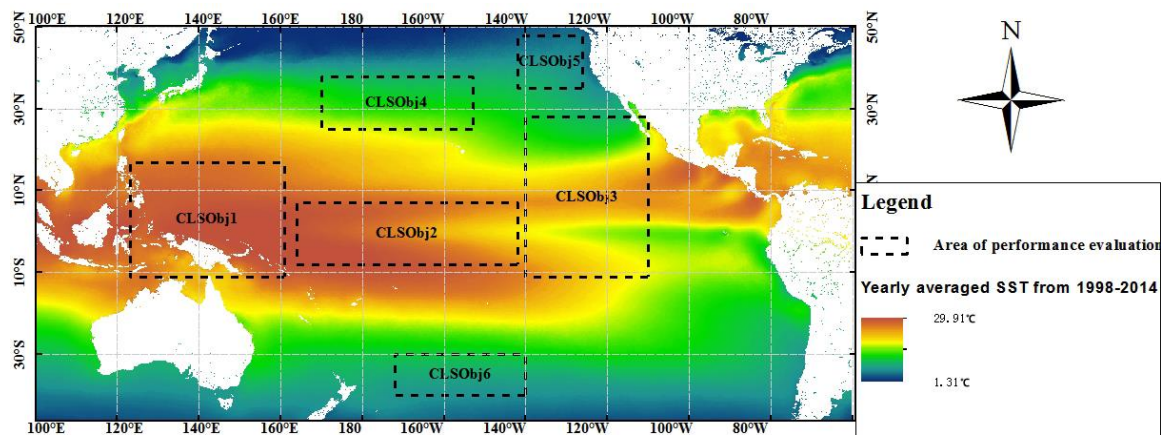
This step generates meaningful cascading patterns from candidates according to the minimum thresholds of the evaluation indicators. Generally speaking, the specified thresholds are defined by users according to their research domains. The evaluation indicators of each cascading pattern, including support and cascading index, are calculated by scanning the database once. If the evaluation indicators satisfy the user-specified thresholds, shown in Equation (18), a cascading pattern is meaningful. The support threshold,  $\sigma_s$ , means the minimum co-occurrence probability of the *Pattern* in the database, while the cascading index,  $\sigma_{CI}$ , means the minimum occurrence probability of each sub-pattern in the database against the *Pattern* occurrence. In this step, the key issue is to scan the database and calculate the evaluation indicators.

$$\left. \begin{array}{l} S(Pattern) \geq \sigma_s \\ CI(Pattern) \geq \sigma_{CI} \end{array} \right\} \quad (18)$$

## 4. Experiments

This section addresses three case studies for M<sup>3</sup>Cap using long time series of remote sensing images. The first experiment analyzes the M<sup>3</sup>Cap's computational complexity from the viewpoints of database scanning and intensive computing, the second illustrates the feasibility and effectiveness of M<sup>3</sup>Cap compared with the derived Apriori algorithm, and the third explores the cascading patterns for marine environmental elements in the Pacific Ocean.

These cases was conducted on long-term marine remote sensing images that showed monthly anomalies of sea surface temperature, named SSTA, monthly anomalies of sea surface chlorophyll-*a*, CHLA, monthly anomalies of sea surface precipitation, SSPA, and monthly anomalies of sea level anomaly, SLAA. The multivariate ENSO index (MEI) was used to identify ENSO events. Table 4 summarizes the datasets used. Marine environmental parameters are ranked into five levels for continuous intervals (−2, −1, 0, 1, and 2) that indicate severe negative change, slight negative change, no change, slight positive change, and severe positive change, respectively. As for the MEI index, −2 indicates strong La Niña events, while +2 indicates strong El Niño events [26]. Monthly image datasets with a spatial resolution of 1° in latitude and longitude from January 1998 to December 2014 for six different regions in the Pacific Ocean, which are denoted as CLSObj1, CLSObj2, CLSObj3, CLSObj4, CLSObj5, and CLSObj6 in Figure 2, are used to test the algorithms. Each region consists of a series of raster lattices, each containing four parameters, SSTA, SSPA, SLAA, and CHLA, with an ENSO index. The experimental hardware environment includes an Intel core i7 CPU at 2.80 GHz, a 500 GB hard disk, and 4.0 GB of memory.



**Figure 2.** Research area. The background colors show the yearly average sea surface temperature (SST) from 1998 to 2014.

**Table 4.** Sources and resolutions of the remote sensing images and multivariate ENSO index (MEI) used in this paper.

	Product	Source	Timespan	Temporal Resolution	Spatial Coverage	Spatial Resolution
1	SST <sup>1</sup>	NOAA/PSD	December 1981–December 2014	Monthly	Global	1° × 1°
2	Chl- <i>a</i> <sup>2</sup>	SeaWifs	September 1997–November 2010	Monthly	Global	9 km × 9 km
3	SSP <sup>3</sup>	MODIS	July 2002–December 2014	Monthly	Global	9 km × 9 km
4	SLA <sup>4</sup>	TRMM	January 1998–December 2014	Monthly	Global	0.25° × 0.25°
5	ENSO <sup>5</sup>	AVISO	January 1993–December 2014	Monthly	Global	0.25° × 0.25°
		MEI	January 1950–December 2014	Monthly	-	-

<sup>1</sup> Obtained from [27] and provided by National Oceanic & Atmospheric Administration Oceanic and Atmospheric Research Earth System Research Laboratory Physical Sciences Division [28]; <sup>2</sup> Obtained from the Sea-viewing Wide Field-of-view Sensor and MODerate-resolution Imaging Spectroradiometer projects and their level three standard mapped images (SMI) [29]; <sup>3</sup> Obtained from Version 7 of the Tropical Rainfall Measuring Mission (TRMM Product 3B43) provided by the Goddard Distributed Active Archive Center; <sup>4</sup> Produced by Ssalto/Duacs and distributed by Archiving, Validation and Interpretation of Satellite Oceanographic data with support from Cnes [30]; <sup>5</sup> Obtained from [31] and provided by National Oceanic & Atmospheric Administration Research Earth System Research Laboratory Physical Sciences Division [32].

#### 4.1. Computational Complexity

The computational complexity of M<sup>3</sup>Cap is classified into two categories—the number of database scans and intensive computing.

Within the M<sup>3</sup>Cap process, scanning a database involves three stages. The first stage builds the mutual information contingency table by scanning the database once. The second stage identifies the one-level cascading patterns from pair-wise related items with quantitative ranks. For each pair-wise related item with each quantitative rank, there is one database scan. For the given quantitative rank,  $R$ , and the total number of items,  $M$ , according to the function, OneLevelCascadingPatternIdentification, the number of database scans is  $(R \times M) \times (R \times (M - 1))$  (i.e.,  $R^2 \times P_M^2$ ) and the computational complexity of identifying the one-level cascading patterns is  $O(R^2 \times M^2)$ . For the third stage, M<sup>3</sup>Cap uses Equation (15) to find a meaningful cascading pattern for each candidate pattern by scanning the database once. Theoretically, there are  $R^2 \times P_M^2 + R^3 \times P_M^3 + \dots + R^M \times M!$  (i.e.,  $O(R^2 \times P_M^2 + R^3 \times P_M^3 + \dots + R^M \times M!)$ ) candidate cascading patterns; hence, the computational complexity of finding a meaningful cascading pattern is  $o(R^2 \times M^2) + o(R^3 \times M^3) + \dots + o(R^M \times M^M)$ . However, in reality, the candidate cascading patterns are reduced substantially in accordance with the information threshold and Properties 2 and 3, which are substantially less than  $R^2 \times P_M^2 + R^3 \times P_M^3 + \dots + R^M \times M!$ .

M<sup>3</sup>Cap involves two stages of intensive computing. One stage obtains the pair-wise related items from a mutual information contingency table. There are  $M \times (M - 1)$  discriminations to determine the related items according to the information threshold; hence, the computational complexity is  $O(M^2)$ . The other stage generates candidate cascading patterns using a recursive linking and pruning

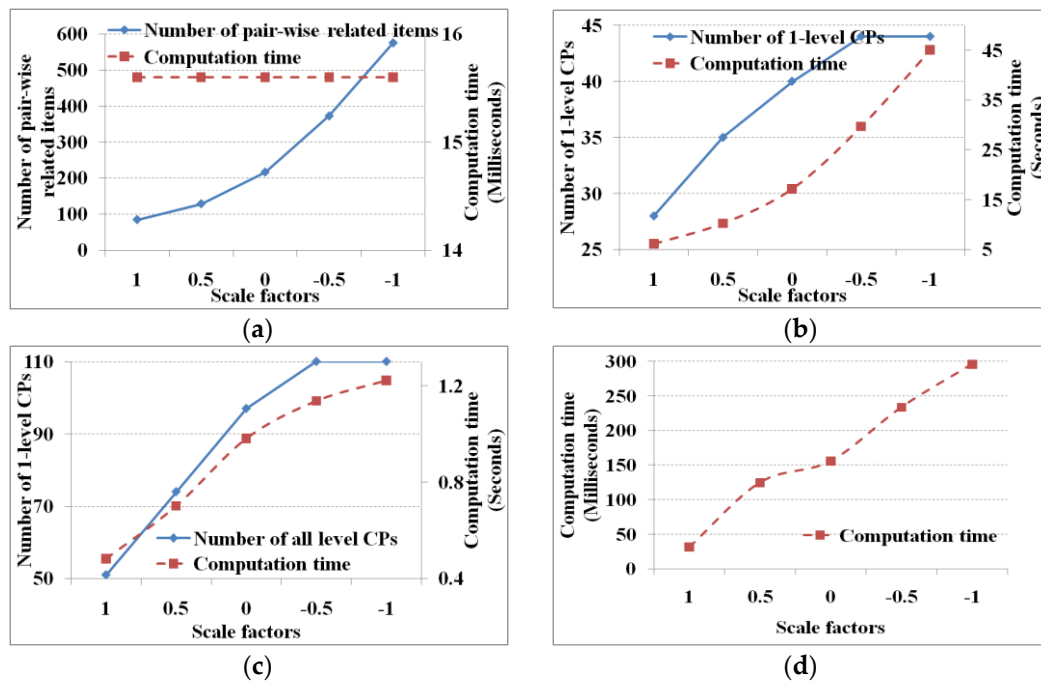
algorithm. In the linking phase, according to Linking Algorithm, the  $(m + 1)$ -level cascading patterns are generated by linking  $m$ -level and one-level patterns, and the number of linkings is  $R^2 \times M^2 \times R^m \times P_M^m$ ; thus, the linking complexity is  $o(R^2 \times M^2 \times R^m \times P_M^m) = o(R^{2+m} \times M^{2+m})$ . In the pruning phase, for each candidate  $(m + 1)$ -level cascading pattern, there are  $(m + 2)$  subpatterns of an  $m$ -level cascading pattern. Each subset needs a calculation to determine whether the  $(m + 1)$ -level cascading pattern is to be removed. The number of discriminations is  $R^2 \times M^2 \times R^m \times P_M^m \times (m + 2)$ ; therefore, the pruning complexity is  $o(m \times R^{2+m} \times M^{2+m})$ .

#### 4.2. Factors Affecting the Performance of M<sup>3</sup>Cap

Given a quantitative rank, three factors affect the performance of M<sup>3</sup>Cap. They are information threshold, evaluation threshold, and database size (i.e., the number of database fields (the derived items)) and records. The information threshold determines how many items are pair-wise related and influences the database base scanning times. The evaluation thresholds are determined by database scanning the number of meaningful cascading patterns from among the candidates. The database fields determine the database scanning time, and the database records determine the computation time for database scanning.

##### 4.2.1. Information Threshold

The mutual information threshold determines the initial pair-wise related items and then obtains cascading patterns. According to Equation (14), the threshold is calculated through the 1.0 mean plus  $\lambda$  standard deviation of the items' mutual information. In this experiment, the support and cascading index thresholds are set to 5% and 75%, respectively, and  $\lambda$  is 1.0, 0.5, 0.0,  $-0.5$ , and  $-1.0$ . Figure 3 shows the performance under mutual information thresholds with different  $\lambda$  values.



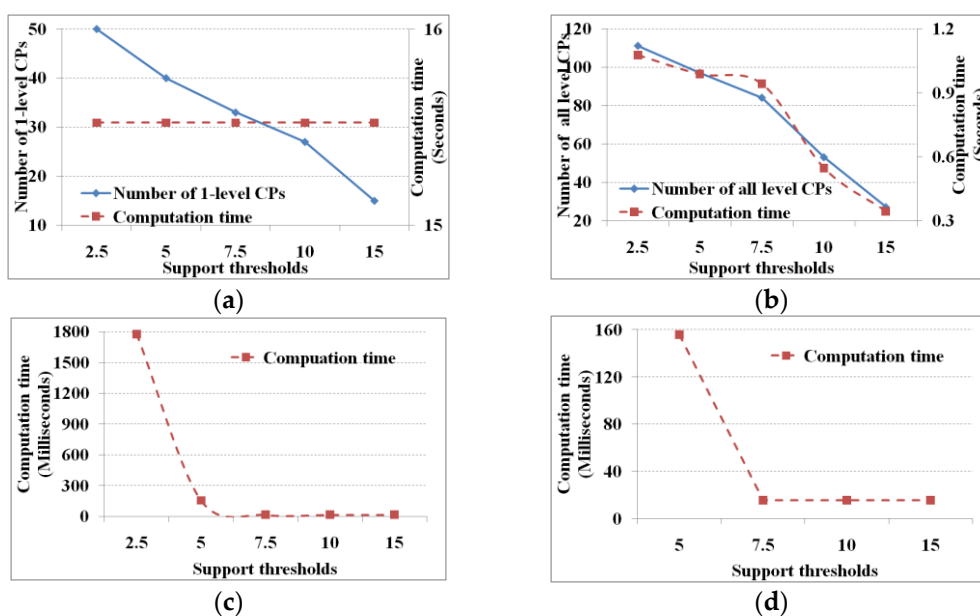
**Figure 3.** Performance under different scale factors of mutual information threshold. (a) Number of pair-wise related items and computation time; (b) Number of one-level cascading patterns and computation time; (c) Total number of cascading patterns and computation time; (d) Computation time of linking and pruning.

Given the support and cascading index thresholds, the smaller the mutual information threshold, the more the pair-wise related items, and then the more one-level and total cascading

patterns. Regarding the computation time, to determine which items are related from the mutual information contingency table, the number of discriminations to obtain the related items is fixed (e.g.,  $25 \times 24$ ). Thus the computation time is fixed (i.e., 15.6 ms) whether  $\lambda$  is large or small (Figure 3a). The computation time to find the one-level and total cascading patterns is determined by the number of database scans (i.e., the number of pair-wise related items and candidate cascading patterns). Hence, the smaller the mutual information threshold, the greater the computation time for finding one-level and total cascading patterns (Figure 3b,c), and vice versa. Similarly, the number of candidate cascading patterns determines the number of linkings and prunings; hence, the smaller the mutual information threshold, the greater the computation time of linking and pruning (Figure 3d). If the database is not scanned, then this computation time is reduced substantially.

#### 4.2.2. Evaluation Thresholds

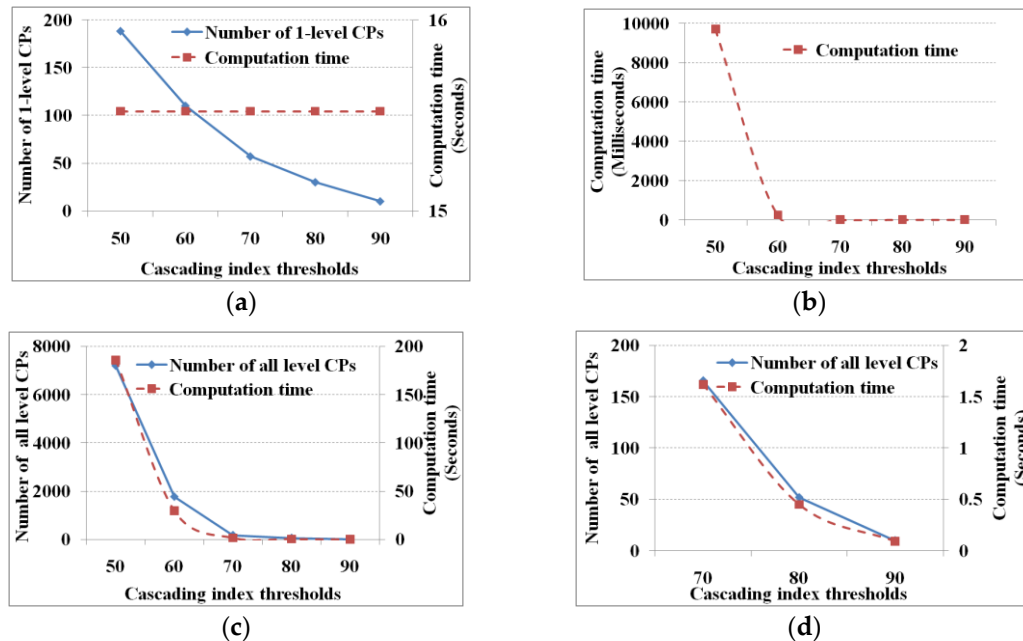
The evaluation thresholds include a support and a cascading index threshold, which are largely defined according to user requirements and domain. M<sup>3</sup>Cap involves two stages in which it uses the evaluation thresholds to find cascading patterns. One stage identifies one-level cascading patterns from pair-wise related items, while the other stage generates meaningful cascading patterns from all the candidates. In this experiment, 25 items within six regions with 204 records are used to test the algorithm; the scale factor of mutual information threshold is set to the mean value (i.e.,  $\lambda = 0$ ) and the support and cascading index thresholds are set to 5% and 75%, respectively. Figures 4 and 5 show the performance under different support and cascading index thresholds, respectively.



**Figure 4.** Performance under different support thresholds. (a) Number of one-level cascading patterns and computation time; (b) Total number of cascading patterns and computation time; (c) Computation time of linking and pruning; (d) Partial enlargement of (c).

From Figures 4 and 5, we know that the lower the evaluation thresholds, the greater are the number of cascading patterns, and vice versa, and the computation time of finding all cascading patterns decreases with increasing thresholds. However, the computation time to find one-level cascading patterns is fixed (i.e., 15.6 s) whether the thresholds are high or low. Given a mutual information threshold, we will get a fixed number of pair-wise related items and database scanning times; hence, the computation time of finding the one-level cascading patterns is fixed. As for the computation time of the linking and pruning function, theoretically it is expected to decrease along with increasing evaluation thresholds. However, when the evaluation thresholds reach a specific value (support and

cascading index thresholds of 7.5% and 70%, respectively) the computation time does not change. When the threshold reaches the specific value, the linking and pruning times are too small to show changes in computation time. That is to say, when the threshold reaches the specified value, the computation time is shown as a computational unit (i.e., 15.6 ms on our test computer).



**Figure 5.** Performance under different cascading index thresholds. (a) Number of one-level cascading patterns and computation time; (b) Computation time of linking and pruning; (c) Total number of cascading patterns and computation time; (d) Partial enlargement of (c).

#### 4.2.3. Database Size

The database size consists of the record and field number. The former represents the number of samples, and the latter represents the derived items. In this experiment, the support and cascading index thresholds are set to 5% and 60%, respectively, and  $\lambda$  is set to zero. Regarding the database records, 1, 10, 100, 1000, and 10,000 copies of 204 records (January 1998 to December 2014) with 25 items (six regions with four parameters and ENSO index) from remote sensing images are produced. As we are mining with duplications, we get similar results with 216 pair-wise related items, 40 one-level cascading patterns, and 97 total cascading patterns. Figure 6 shows the performance of M<sup>3</sup>Cap under different record numbers (i.e.,  $204 \times 10^0$ ,  $204 \times 10^1$ ,  $204 \times 10^2$ ,  $204 \times 10^3$ , and  $204 \times 10^4$ , respectively).

Figure 6 shows that maximum computation time occurs in finding the one-level cascading patterns then in finding all cascading patterns and pair-wise related items in order and finally in linking and pruning. The computation time is dominated primarily by the number of database scans, which is determined by the number of database records. In this experiment, there are  $216 \times 25$  database scans to find one-level cascading patterns, 105 to find all cascading patterns, and one to construct a mutual information contingency table and obtain the pair-wise related items. Apart from the effects of the database records, the computation time of linking and pruning is fixed irrespective of a database record size. In addition, we know that except for linking and pruning, the computation time increases exponentially with an increase in the number of database records.

To test the field number on the performance of M<sup>3</sup>Cap, we combine the four marine parameters within each region (i.e., CLSObj1 to CLSObj6 with ENSO index, as shown in Figure 2). Thus, 5, 9, 13, 17, 21, and 25 fields are used (e.g., the five fields are the four marine parameters in the CLSObj1 region with the ENSO index, the nine fields are the four marine parameters in the CLSObj1 and CLSObj2 regions, respectively, with ENSO index, and so on). In this experiment, the support and cascading



index thresholds are set to 5% and 75%, respectively, and  $\lambda$  is set to zero. The results are shown in Figure 7.

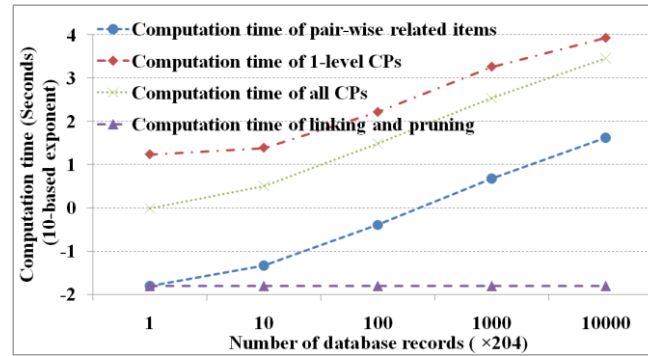


Figure 6. Performance analysis with different record numbers.

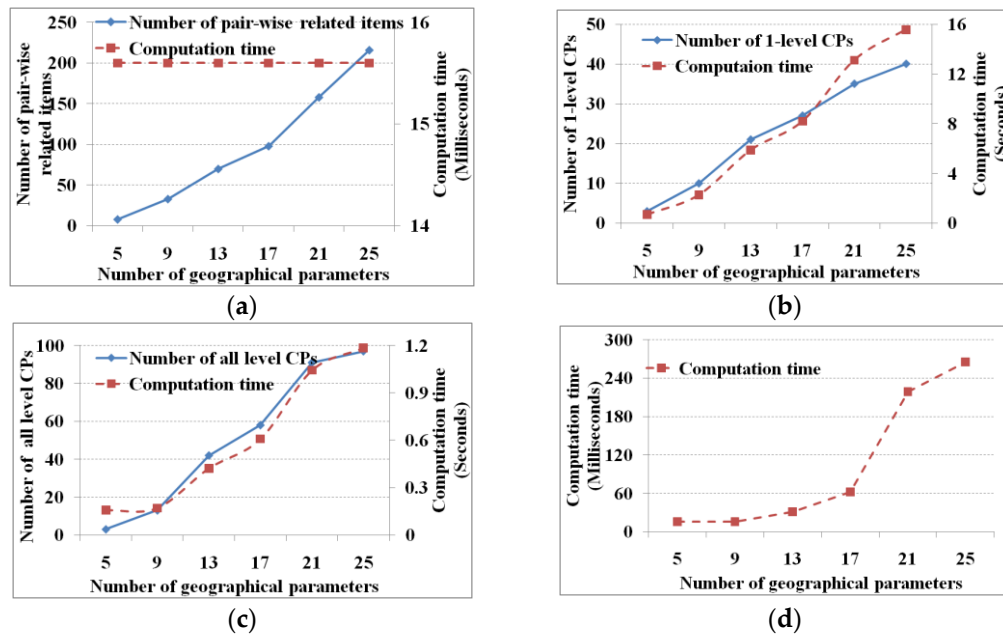


Figure 7. Performance analysis with different field numbers. (a) Number of pair-wise related items and computation time; (b) Number of one-level cascading patterns and computation time; (c) Number of cascading patterns and computation time; (d) Linking and pruning computation time.

Along with the increase in the number of items, we spend more time to obtain the further numbers of one-level and all-level cascading patterns (Figure 7b,c) and to further link candidate cascading patterns and remove pseudo patterns (Figure 7d). Although we obtain more pair-wise related items, the computation time is almost fixed (Figure 7a). The reason lies in that the computation time focuses on discrimination times, which changes from  $5 \times 4$ ,  $9 \times 8$ ,  $13 \times 12$ ,  $17 \times 16$ ,  $21 \times 20$  to  $25 \times 24$ . Comparative to the computational abilities, the computation time for discriminating pair-wise related items gives no significant change.

#### 4.3. Comparisons with the Quantative GSP

We use the generalized sequence pattern, GSP [12], as the baseline for comparison on the efficiency of the M<sup>3</sup>Cap algorithm. The three core steps of GSP are (1) finding the frequent 1-items to generate one-level sequence patterns; (2) linking and pruning to generate candidate patterns; and

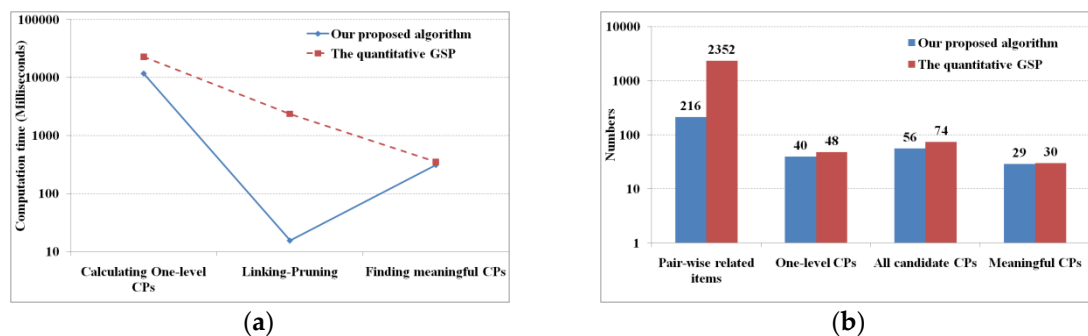
(3) counting candidates to find meaningful sequence patterns. For dealing with the quantitative patterns, the quantitative GSP is derived from the core idea of GSP, and in order to make a fair comparison, the quantitative GSP uses the same discretization strategy, the support and cascading index thresholds as the M<sup>3</sup>Cap.

In this experiment, 25 items within six regions with 204 records are used to test the algorithms, the scale factor of mutual information threshold is set to the mean value (i.e.,  $\lambda = 0$ ), and the support and cascading index thresholds are set to 5% and 75%, respectively. For a clear comparison, both GSP and M<sup>3</sup>Cap are divided into three mining phases (i.e., a calculating of one-level cascading patterns, a recursive of linking and pruning, and a finding of meaningful cascading patterns). Figure 8 shows their computation time and results, and Table 5 shows a comparison of parts of the mined meaningful cascading patterns (only the cascading patterns with a support not less than 15.00%).

**Table 5.** Comparisons between M<sup>3</sup>Cap and generalized sequential pattern (GSP) results.

	No.	Cascading Patterns	S (%)	CI (%)
GSP	1	CHLA-CLSOBJ1[-2] * → SSTA-CLSOBJ1[2]	15.00	93.10
M <sup>3</sup> Cap and GSP	2	SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ3[-2]	23.33	84.00
	3	SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ4[2]	21.67	78.00
	4	SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2]	23.33	85.71
	5	SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ4[2]	22.78	83.67
	6	SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ5[-2]	20.56	75.51
	7	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2]	19.44	92.11
	8	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ3[-2]	18.33	86.84
	9	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ4[2]	16.67	78.95
	10	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ5[-2]	17.22	81.58
	11	SSTA-CLSOBJ5[-2] → SSTA-CLSOBJ4[2]	22.78	80.39
	12	SLAA-CLSOBJ6[2] → SLAA-CLSOBJ1[2]	21.11	77.55
	13	ENSO[-2] → SSTA-CLSOBJ2[-2]	17.78	91.43
	14	ENSO[-2] → SSTA-CLSOBJ3[-2]	18.89	97.14
	15	ENSO[-2] → SSTA-CLSOBJ4[2]	17.22	88.57
	16	ENSO[-2] → SSTA-CLSOBJ5[-2]	15.56	80.00
	17	SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ4[2]	19.44	83.33
	18	SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ4[2]	19.44	83.33
	19	SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ5[-2] → SSTA-CLSOBJ4[2]	17.22	75.51
	20	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ3[-2]	17.78	91.43
	21	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ4[2]	15.00	77.14
	22	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2]	17.78	86.84
	23	SLAA-CLSOBJ3[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ5[-2]	15.56	84.85
	24	ENSO[-2] → SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ3[-2]	17.78	91.43
	25	ENSO[-2] → SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ4[2]	15.56	87.50
	26	ENSO[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2]	17.78	94.12
	27	ENSO[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ4[2]	16.67	88.24
	28	ENSO[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ5[-2]	15.00	79.41
	29	ENSO[-2] → SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ4[2]	15.56	87.50
	30	ENSO[-2] → SSTA-CLSOBJ3[-2] → SSTA-CLSOBJ2[-2] → SSTA-CLSOBJ4[2]	15.56	87.50

\* CLSOBJ with a number after the marine parameters indicates the research area, the number within the square bracket indicates the variation degree (e.g., CHLA-CLSOBJ1[-2] means that in region CLSOBJ1, CHLA shows an abnormal decrease compared with the monthly averaged).



**Figure 8.** Comparisons of the quantitative GSP and our proposed algorithm. (a) Computation time during the different mining phases; (b) Number of pair-wise related items, one-level cascading patterns, all candidates of cascading patterns, and the meaningful cascading patterns.

In the quantitative GSP algorithm, a calculating of one-level CPs includes the steps of finding frequent 1-items and of generating two dimensional association patterns (i.e., one-level cascading patterns) while in M<sup>3</sup>Cap, this includes the steps of identifying pair-wise related items and of generating one-level cascading patterns. In this step, the M<sup>3</sup>Cap is two times faster than GSP, i.e., the former takes 11.6 s, while the latter dose 22.9 s (Figure 8a). The reason lies in that M<sup>3</sup>Cap scans database one time to calculate the items' mutual information, and then generates the pair-wise related items, while GSP scans the database once for each item to identify it frequent or not, and then generates the pair-wise related items. From pair-wise related items to one-level cascading patterns, both GSP and M<sup>3</sup>Cap scan the database once for each pair-wise related item, however, the former deals with 2352 pieces of pair-wise related items, while the latter does 216 pieces. In the recursive of the linking and pruning phase, both GSP and M<sup>3</sup>Cap generate all the candidates of cascading patterns. The former will scan database once for each candidate cascading pattern, while M<sup>3</sup>Cap will not, therefore, M<sup>3</sup>Cap computes all candidates of cascading patterns up to two orders of magnitude faster than GSP. In the final step, M<sup>3</sup>Cap and GSP have the similar processing method of finding meaningful cascading patterns by scanning the database once for each candidate, thus, their computing efficiencies have no significant differences.

Table 5 shows that the mined results from M<sup>3</sup>Cap and GSP are the same, except that GSP finds an additional patterns, "CHLA-CLSOBJ1[-1] → SSTA-CLSOBJ1[2], 15.00%, 93.10%". The reason lies in the selection of the mutual information threshold. The standard mutual information from CHLA-CLSOBJ1 to SSTA-CLSOBJ1 is 0.100, while the mean standard mutual information for all the items, the information threshold, is 0.104. That is to say, the candidate "CHLA-CLSOBJ1 → SSTA-CLSOBJ1" is missed when the pair-wise related items are found at the first step, so the pattern is not found by the M<sup>3</sup>Cap algorithm.

#### 4.4. Marine Cascading Patterns in the Pacific Ocean

Supposed the cascading patterns in Table 5 are meaningful (fitting to the minimum support and cascading index thresholds), some well-known interrelationships within the Pacific Ocean are obtained: the variation of SSTA, SLAA, or co-variations between SSTA and SLAA mostly are dominated by ENSO, Pacific Decadal Oscillation, and Pacific-North American patterns, thus, SSTA and SLAA are more interrelated than other marine environmental parameters (No.2 to No.12) [33]. When La Niña events occur, the westward Rossby reaches the central Pacific Ocean, which pushes the warm pool at the surface toward the west. At the same time, the North Pacific Current flows eastward through the middle of the northern subtropical Pacific Ocean, resulting in mean SST increases. So, SSTA in regions CLSOBJ2, CLSOBJ3 and CLSOBJ4 is a response to a La Niña event (No.13 to No.15) [2]. In addition, we can also discern much more quantitative information among marine environmental parameters, and identify lesser-known patterns from multiple marine environmental parameters, such as in No.16, meaning that an abnormal decrease of SSTA in region CLSOBJ4 may be a response to a La Niña event. From two- to more- level cascading patterns (No.17 to No.30) we also find some relationships in order in different regions. Such patterns may be new to Earth scientists.

## 5. Discussion and Conclusions

This paper proposes a novel algorithm—M<sup>3</sup>Cap—for mining cascading patterns and tests the algorithm using remote sensing image datasets. By delving into the mutual information to extract the initial pair-wise related items and using a recursive linking and pruning function without database scanning, M<sup>3</sup>Cap reduces the number of database scans substantially and improves computational efficiency.

Although our proposed M<sup>3</sup>Cap has a workflow similar to the previously proposed MIQarma [24], the two approaches are essentially different. M<sup>3</sup>Cap focuses on cascading characteristics, while MIQarma deals with association characteristics; hence, they differ in the key issue of linking and pruning. With no database scanning during the linking and pruning, the efficiency of M<sup>3</sup>Cap increases.

In addition, the core idea of M<sup>3</sup>Cap is based on quantitative GSP [12], but the two algorithms differ in initial pair-wise related item (frequent two-item) extraction, linking and pruning, and evaluation indicators.

Analysis of computational complexity shows that M<sup>3</sup>Cap involves four stages—extraction of pair-wise related items, recursive linking and pruning to obtain all of the candidate cascading patterns, extraction of one-level cascading patterns, and generation of meaningful cascading patterns. The first two stages involve intensive computing and are dependent on the number of derived items and on linking and pruning. The other two are dependent on the number of database scans and the time for scanning the database once. Theoretically, the second stage involves the most complex intensive computing; however, with no database scanning, its computational complexity is simplest.

Regarding database scanning times, the mutual information threshold is one of the most important factors. Given the number of derived items, a smaller mutual information threshold will result in more initial related pair-wise items and more database scans, and vice versa. Thus, the key issue is to determine the mutual information threshold. M<sup>3</sup>Cap combines the mean and standard deviation of the items' mutual information, which is commonly used to reveal highly related items [21]. Moreover, many experiments show that when a mutual information threshold is between a value of the mean subtracting one standard deviation and a value of the mean adding one standard deviation (i.e., a scalar factor is between  $-1$  and  $1$ ), we will obtain approximate cascading patterns as typical mining algorithms. In addition, the evaluation thresholds, a support threshold and a cascading index, affect the number of database scans. The smaller the evaluation indicators, the more the one-level cascading patterns, then the more other-level patterns, then the more times of database scanning, and vice versa. In this paper, the evaluation thresholds are defined according to users' requirements and domain. Experiments with remote sensing image datasets showed that the computation time of database scanning once is mostly dominated by the number of database records. A greater number of database records results in a greater computation time for database scanning. However, the linking and pruning computation time with different thresholds, shown in Figures 4c and 5b, contradict the theoretical analysis. Theoretically, as evaluation thresholds increase, the candidate cascading patterns decrease and the linking and pruning time decreases along with its computation time. However, when the support threshold and cascading index reach 7.5% and 70%, respectively, their computation times do not decrease because the linking and pruning time is too short to show changes in the computation time. Similarly, given a mutual information threshold, the more the derived items, the more the discrimination criterion is applied to obtain pair-wise related items and the more computation time is then required. However, the computation time does not change, as Figure 7a shows. The number of derived items changes from 5 to 25, and the number of applications of the discrimination criterion changes from  $5 \times 4$  to  $25 \times 24$  correspondingly. The limited discrimination criterion times are not sufficient to be able to show changes in computation time.

In summary, faced with the challenge of finding cascading characteristics, we proposed the M<sup>3</sup>Cap method and tested it using real remote sensing image datasets. The main conclusions of our study are the following:

1. We designed the M<sup>3</sup>Cap workflow and discussed M<sup>3</sup>Cap computational complexities by absorbing mutual information. According to the two categories of complexity—intensive computing and database scans—four key steps were presented.
2. Two stages dominate intensive computing, obtaining pair-wise related items from a mutual information contingency table according to the discrimination criterion, which is dominated by the number of derived items, and generating all of the candidate cascading patterns from one-level patterns by recursive linking and pruning. Having no database scanning substantially reduces computational complexity.
3. The mutual information and evaluation thresholds and the database size are the main factors affecting the number of database scans. The mutual information threshold mainly dominates the initial pair-wise related items and affects the number of candidate cascading patterns indirectly.

The evaluation thresholds dominate the time of finding the meaningful cascading patterns. The number of database fields (i.e., the number of derived items) also affects the initial pair-wise related items, then the one-level and all cascading patterns, and the number of database records dominates the time of database scanning once.

4. Experiments with real remote sensing image datasets covering the Pacific Ocean were conducted to demonstrate the feasibility and efficiency of M<sup>3</sup>Cap.

**Acknowledgments:** This research was supported by National key research and development program of China (No.2016YFA0600304), and by the National Natural Science Foundation of China (No. 41371385 and No. 41476154), and by the Youth Innovation Promotion Association of Chinese Academy of Science (No. 2013113).

**Author Contributions:** Cunjin Xue led the research work, proposed the idea and structure of this manuscript, wrote the introduction and the mining algorithm sections, and integrated the contributions from all authors as a whole. Jingyi LIU and Xiaohong LI contributed the M<sup>3</sup>Cap algorithm and experiment, and collaborated with Cunjin Xue in writing the experiment section. Qing Dong proposed the idea of this manuscript and conceived and designed the experiments, and collaborated with Cunjin Xue in writing the results and discussion sections.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Blanchart, P.; Ferecatu, M.; Cui, S.Y.; Datcu, M. Pattern retrieval in large image databases using multi scale coarse-to-fine cascaded active learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1127–1141. [[CrossRef](#)]
2. McPhaden, M.J.; Zebiak, S.E.; Glantz, M.H. ENSO as an integrating concept in earth science. *Science* **2006**, *314*, 1740–1745. [[CrossRef](#)] [[PubMed](#)]
3. Qiu, J.N.; Wang, Z.Q.; Ye, X.; Liu, L.; Dong, L. Modeling method of cascading crisis events based on merging Bayesian Network. *Decis. Support Syst.* **2014**, *62*, 94–105. [[CrossRef](#)]
4. Agrawal, R.; Srikant, R. Mining sequential patterns. In Proceedings of the 1995 International Conference on Data Engineering (ICDE'95), Taipei, Taiwan, 19 March 1995.
5. Mooney, C.H.; Roddick, J.F. Sequential pattern mining approaches and algorithms. *ACM Comput. Surv.* **2013**, *45*, 1–39. [[CrossRef](#)]
6. Lee, A.J.T.; Chen, Y.A.; Ip, W.C. Mining frequent trajectory patterns in spatial-temporal databases. *Inf. Sci.* **2009**, *179*, 2218–2231. [[CrossRef](#)]
7. Lu, E.H.C.; Tseng, V.S.; Yu, P.S. Mining cluster-based temporal mobile sequential patterns in location-based service environments. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 914–927. [[CrossRef](#)]
8. McGuire, M.P.; Janeja, V.P.; Gangopadhyay, A. Mining trajectories of moving dynamic spatio-temporal regions in sensor datasets. *Data Min. Knowl. Discov.* **2014**, *28*, 961–1003. [[CrossRef](#)]
9. Huang, Y.; Zhang, L.Q.; Zhang, P.S. A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 433–448. [[CrossRef](#)]
10. Nakagaito, F.; Ozaki, T.; Ohkawa, T. Discovery of quantitative sequential patterns from event sequences. In Proceedings of the IEEE International Conference on Data Mining Workshops, Washington, DC, USA, 6 December 2009; pp. 31–36.
11. Mohan, P.; Shekhar, S.; Shine, J.A.; Rogers, J.P. Cascading spatio-temporal pattern discovery. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 1977–1992. [[CrossRef](#)]
12. Srikant, R.; Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. In Proceeding of the 5th International Conference on Extending Database Technology (EDBT'96), Avignon, France, 25–29 March 1996; pp. 3–17.
13. Zaki, M. SPADE: An efficient algorithm for mining frequent sequences. *Mach. Learn.* **2001**, *42*, 31–60. [[CrossRef](#)]
14. Febrer-Hernández, J.K.; Hernández-Palancar, J.; Hernández-León, R.; Feregrino-Urbe, C. SPaMi-FTS: An efficient algorithm for mining frequent sequential patterns. In Proceedings of the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Puerto Vallarta, Mexico, 2–5 November 2014.



15. Han, J.W.; Cheng, H.; Xin, D.; Yan, Y.F. Frequent pattern mining: Current status and future directions. *Data Min. Knowl. Discov.* **2007**, *15*, 55–86. [[CrossRef](#)]
16. Han, J.W.; Pei, J.; Mortazavi-Asl, B.; Chen, Q.; Dayal, U.; Hsu, M. Frequent pattern-projected sequential pattern mining. In *Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery in Databases (KDD'00)*, Boston, MA, USA, 20–23 August 2000.
17. Pei, J.; Han, J.; Mortazavi-Asl, B.; Wang, J.; Pinto, H.; Chen, Q.; Dayal, U.; Hsu, M.C. Mining sequential patterns by pattern-growth: The prefix span approach. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1424–1440.
18. Hu, Y.H.; Wu, F.; Liao, Y.J. An efficient tree-based algorithm for mining sequential patterns with multiple minimum supports. *J. Syst. Softw.* **2013**, *86*, 1224–1238. [[CrossRef](#)]
19. Fabrègue, M.; Braud, A.; Bringay, S.; Ber, F.L.; Teisseire, M. Mining closed partially ordered patterns, a new optimized algorithm. *Knowl.-Based Syst.* **2015**, *79*, 68–79. [[CrossRef](#)]
20. Batina, L.; Gierlichs, B.; Prouff, E.; Rivain, M.; Standaert, F.X.; Nicolas, V.C. Mutual information analysis: A comprehensive study. *J. Cryptol.* **2011**, *24*, 269–291. [[CrossRef](#)]
21. Ke, Y.P.; Cheng, J.; Ng, W. An information-theoretic approach to quantitative association rule mining. *Knowl. Inf. Syst.* **2008**, *16*, 213–244. [[CrossRef](#)]
22. Sy, B.K. Discovering association patterns based on mutual information. In *Proceeding of the 3rd International Conference (MLDM 2003)*, Leipzig, Germany, 5–7 July 2003; pp. 369–378.
23. Ke, Y.P.; Cheng, J.; Ng, W. Correlated pattern mining in quantitative databases. *ACM Trans. Database Syst.* **2008**, *33*, 1–44. [[CrossRef](#)]
24. Xue, C.J.; Song, W.J.; Qin, L.J.; Dong, Q. A normalized-mutual-information-based mining method for marine abnormal association rules. *Comput. Geosci.* **2015**, *76*, 121–129.
25. Geng, L.; Hamilton, H.J. Interestingness measures for data mining: A survey. *ACM Comput. Surv.* **2006**, *38*, 1–32. [[CrossRef](#)]
26. Xue, C.J.; Song, W.J.; Qin, L.J.; Dong, Q.; Wen, X.Y. A spatiotemporal mining framework for abnormal association patterns in marine environments with a time series of remote sensing images. *Int. J. Appl. Earth Obs. Geoinform.* **2015**, *38*, 105–114. [[CrossRef](#)]
27. ESRL. Available online: <http://www.esrl.noaa.gov/psd/> (accessed on 24 May 2016).
28. Reynolds, R.W.; Rayner, N.A.; Smith, T.M.; Stokes, D.C.; Wang, W. An improved in situ and satellite SST analysis for climate. *J. Clim.* **2002**, *15*, 1609–1625. [[CrossRef](#)]
29. Hooker, S.B.; McClain, C.R. The calibration and validation of SeaWiFS data. *Prog. Oceanogr.* **2000**, *45*, 427–465. [[CrossRef](#)]
30. AVISO. Available online: <http://www.aviso.oceanobs.com/duacs> (accessed on 24 May 2016).
31. NOAA. Available online: <http://www.esrl.noaa.gov/psd/enso/mei/> (accessed on 24 May 2016).
32. Wolter, K.; Timlin, M.S. El Niño/Southern Oscillation behavior since 1871 as diagnosed in an extended multivariate ENSO index (MEI.ext). *Int. J. Climatol.* **2011**, *31*, 1074–1087. [[CrossRef](#)]
33. Xue, C.J.; Dong, Q.; Fan, X. Spatiotemporal association patterns of multiple parameters in the northwestern Pacific Ocean and their relationships with ENSO. *Int. J. Remote Sens.* **2014**, *35*, 4467–4483. [[CrossRef](#)]

