Towards Unexpected Sequential Patterns

Haoyuan Li

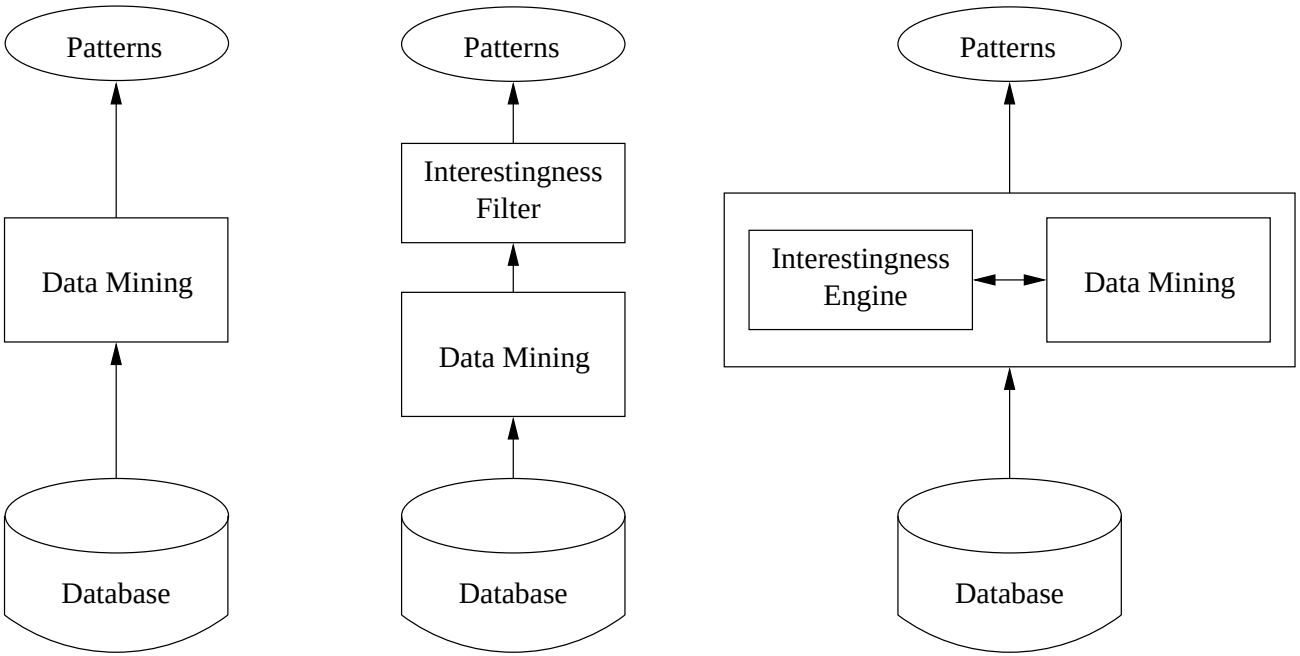
LGI2P - École des Mines d'Alès
LIRMM - Université Montpellier II

3 July 2007

Outline

Outline

Frequent Pattern Mining and Interestingness Measure



1. Finding all frequent patterns
2. Finding all interesting frequent patterns
3. Finding all interesting patterns (even not frequent)

Mining Sequential Patterns

Computational Task

- Given a data set of sequences (or a transactional database), find maximal sequences that satisfy the given threshold *minimal support* σ.

Example

- With $\sigma = 0.5$, we find that 60% of customers who bought a iBook will buy an iPhone later.

Problem

Which one is more interesting?

1. 60% of customers who buy a iBook will buy an iPhone later.
2. 2% of customers who buy a iBook will buy a Windows Mobile later.

Unexpected Sequence Mining

- A new field in the sequence mining domain
- Depends on domain knowledge and semantics
- Widely applicable

Outline

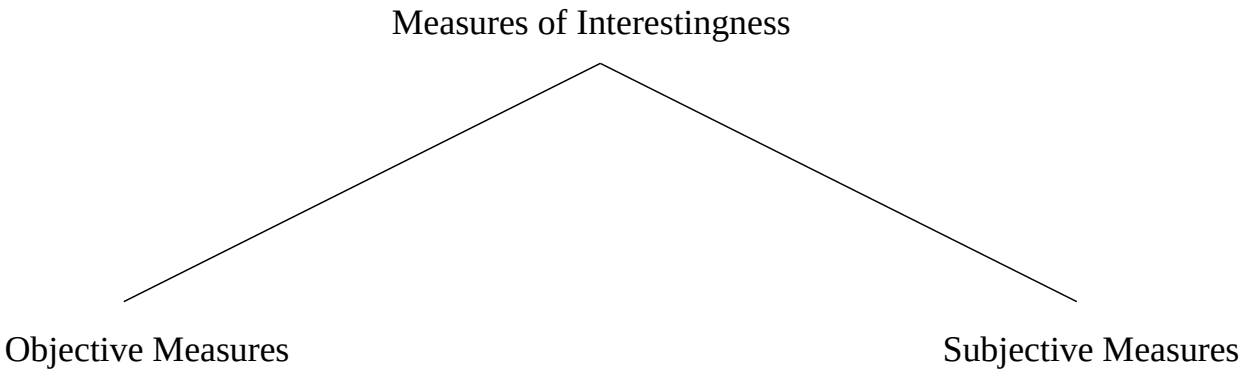
Interestingness Measures for Data Mining

References

- Abraham Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In KDD, pages 275–281, 1995.
- Abraham Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. IEEE Trans. Knowl. Data Eng., 8(6), 1996.

- Ken McGarry. A survey of interestingness measures for knowledge discovery. Knowl. Eng. Rev., 20(1):39–61, 2005.

Classification of Interestingness Measures



Objective Measures

Principal

- Depend on the structure of patterns.
- Based on statistics or probability approaches.

Example

- Association rule: $iBook \rightarrow iPhone$
- Support: $\frac{|iBook \cup iPhone|}{|All\ \ Purchases|}$
- Confidence: $\frac{|iBook \cup iPhone|}{|iBook|}$

Subjective Measures

Principal

- Depend on the class of users.
- Based on knowledge/belief systems and semantics.

Example

- Domain knowledge: $iPhone \Rightarrow \neg Windows\ Mobile$
- User belief: $iBook \rightarrow iPhone$
- Unexpected rule: $iBook \rightarrow Windows\ Mobile$

Belief-Driven Unexpectedness

Definition

- A *belief system* is a set of domain knowledge related user defined constraints.
- The *unexpectedness* is the violations between patterns and beliefs.

Classification of Beliefs

- Hard belief
- Soft belief

Hard Belief

Principal

- A *hard belief* cannot be changed with new evidences.
- Violations mean unexpected data.

Example

- Hard belief: $STOP \rightarrow Car\ stops$
- New evidence: $STOP \rightarrow Car\ passes$
- Action: Alarm?

Soft Belief

Principal

- A *soft belief* is measured by a *degree* which can be changed with new evidences.
- The change of degree is performed by user-defined criteria.
- The interestingness is measured by the change of the degree.

Example

- Soft belief: $iBook \rightarrow iPhone$, $degree = 0.9$
- New evidence: $iBook \rightarrow Windows\ Mobile$, $confidence = 0.3$
- Action: $iBook \rightarrow iPhone$, $degree = 0.8$

Unexpected Association Rules

References

- Balaji Padmanabhan and Alexander Tuzhilin. Unexpectedness as a measure of interestingness in knowledge discovery. Decision Support Systems, 27:303–318, 1999.
- Balaji Padmanabhan and Alexander Tuzhilin. On characterization and discovery of minimal unexpected patterns in rule discovery. IEEE Trans. Knowl. Data Eng., 18(2):202–216, 2006.

Unexpected Association Rule

Definition

- Belief $b = X \rightarrow Y$ (given by domain experts).
- Rule $p = A \rightarrow B$ is unexpected to b:
  1. $B = CONTR(Y)$ (logical contradiction in semantics);
  2. If $A \cup X \rightarrow Y$ does not hold given confidence/support;
  3. If $A \cup X \rightarrow B$ holds given confidence/support.

The ZoominUR Algorithm

Input : $\mathcal{B}$, $\mathcal{D}$, σ and δ

Output: Itemsets for each belief $b \in \mathcal{B}$

1 **foreach** $b : (X \rightarrow Y) \in \mathcal{B}$ **do**
2 | Find all itemsets $\mathcal{I}_C = \{\mathcal{I} | X \cup CONTR(Y) \subset \mathcal{I}\}$ from $\mathcal{D}$ with respect to σ, by the *a priori* approach;
3 **end**
4 **foreach** $\mathcal{I} \in \mathcal{I}_C$ **do**
5 | **foreach** $b : (X \rightarrow Y) \in \mathcal{B}$ **do**
6 | | $a := CONTR(b)$;
7 | | Output all rules $(x \setminus a) \rightarrow a, x \subseteq \mathcal{I}$ with respect to δ;
8 | **end**
9 **end**

Outline

Formal Models of Sequence

Item, Itemset and Sequence

Definition (Item)

Given a set of distinct attributes, an *item* is an attribute, denoted by $\mathbf{i}$. We use $A, B, C, \ldots$ for describing individual items.

Definition (Itemset)

An *itemset* $\mathcal{I} = (\mathbf{i}_1 \mathbf{i}_2 \ldots \mathbf{i}_m)$ is an unordered collection of items.

Definition (Sequence)

A *sequence* $\mathbf{s} = \langle \mathcal{I}_1 \mathcal{I}_2 \ldots \mathcal{I}_k \rangle$ is an ordered list of itemsets.

Segment of Sequence

Definition

$$\mathbf{s} = \langle \mathcal{I}_1 \mathcal{I}_2 \ldots \mathcal{I}_m \rangle$$
$$\mathbf{s}' = \langle \mathcal{I}'_1 \mathcal{I}'_2 \ldots \mathcal{I}'_n \rangle$$

If there exist integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that

$$\mathcal{I}_1 = \mathcal{I}'_{i_1}, \mathcal{I}_2 = \mathcal{I}'_{i_2}, \ldots, \mathcal{I}_m = \mathcal{I}'_{i_m},$$

then sequence $\mathbf{s}$ is a *segment* of sequence $\mathbf{s}'$.

Inclusion of Sequences

Definition

$$\mathbf{s} = \langle \mathcal{I}_1 \mathcal{I}_2 \ldots \mathcal{I}_m \rangle$$
$$\mathbf{s}' = \langle \mathcal{I}'_1 \mathcal{I}'_2 \ldots \mathcal{I}'_n \rangle$$

If there exist integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that

$$\mathcal{I}_1 \subseteq \mathcal{I}'_{i_1}, \mathcal{I}_2 \subseteq \mathcal{I}'_{i_2}, \ldots, \mathcal{I}_m \subseteq \mathcal{I}'_{i_m},$$

then sequence $\mathbf{s}$ *is included in* sequence $\mathbf{s}'$, denoted by

$$\mathbf{s} \sqsubseteq \mathbf{s}'.$$

Bordered Inclusion of Sequences

Definition

$$\mathbf{s} = \langle \mathcal{I}_1 \mathcal{I}_2 \ldots \mathcal{I}_m \rangle$$
$$\mathbf{s}' = \langle \mathcal{I}'_1 \mathcal{I}'_2 \ldots \mathcal{I}'_n \rangle$$

If there exist integers $1 < i_2 < \ldots < i_{m-1} < n$ such that

$$\mathcal{I}_1 \subseteq \mathcal{I}'_1, \mathcal{I}_2 \subseteq \mathcal{I}'_{i_2}, \ldots, \mathcal{I}_{m-1} \subseteq \mathcal{I}'_{m-1}, \mathcal{I}_m \subseteq \mathcal{I}'_n$$

then sequence $\mathbf{s}$ *is bordered included in* sequence $\mathbf{s}'$, denoted by

$$\mathbf{s} \sqsubseteq_{\perp}^{\top} \mathbf{s}'.$$

Example of Segment and Inclusions

Example

$$\mathbf{s}_1 = \langle (A)(B)(C) \rangle$$
$$\mathbf{s}_2 = \langle (AD)(EF)(BD)(EF)(CD) \rangle$$
$$\mathbf{s}_3 = \langle (EF)(AD)(EF)(BD)(EF)(CD)(EF) \rangle$$

- $\mathbf{s}_2$ is a segment of $\mathbf{s}_3$
- $\mathbf{s}_1 \sqsubseteq \mathbf{s}_2$, $\mathbf{s}_1 \sqsubseteq \mathbf{s}_3$
- $\mathbf{s}_1 \sqsubseteq_{\perp}^{\top} \mathbf{s}_2$, $\mathbf{s}_1 \not\sqsubseteq_{\perp}^{\top} \mathbf{s}_3$

Ordered Relations Between Subsequences in a Sequence

Definition ($\mapsto, \mapsto^*, \mapsto^n$)

- $\mathbf{s}_j \mapsto \mathbf{s}_k$: $\mathbf{s}_j$ is **directly followed by** $\mathbf{s}_k$.
- $\mathbf{s}_j \mapsto^* \mathbf{s}_k$: $\mathbf{s}_j$ is **followed by** $\mathbf{s}_k$.
- $\mathbf{s}_j \mapsto^n \mathbf{s}_k$: $\mathbf{s}_j$ is followed by $\mathbf{s}_k$ and between $\mathbf{s}_j$ and $\mathbf{s}_k$ there **must be** n **itemsets**.

Definition ($\not\mapsto, \not\mapsto^*, \not\mapsto^n$)

- $\mathbf{s}_j \not\mapsto \mathbf{s}_k$: $\mathbf{s}_j$ is **not directly followed by** $\mathbf{s}_k$.
- $\mathbf{s}_j \not\mapsto^* \mathbf{s}_k$: $\mathbf{l}_j$ is **not followed by** $\mathbf{s}_k$.
- $\mathbf{s}_j \not\mapsto^n \mathbf{s}_k$: if $\mathbf{s}_j$ is followed by $\mathbf{s}_k$, then between $\mathbf{s}_j$ and $\mathbf{s}_k$ there **must not be** n **itemsets**.

Example of Ordered Relations

Example

$$\mathbf{s} = \langle (A)(BEF)(ACD)(CEF)(AD) \rangle$$
$$\mathbf{s}_1 = \langle (A)(EF) \rangle, \mathbf{s}_2 = \langle (A)(D) \rangle$$
$$\mathbf{s}_3 = \langle (F)(A) \rangle, \mathbf{s}_4 = \langle (D) \rangle, \mathbf{s}_5 = \langle (A)(B) \rangle$$

- $\mathbf{s}_1 \mapsto \mathbf{s}_1$, $\mathbf{s}_1 \mapsto \mathbf{s}_4$, $\mathbf{s}_1 \mapsto^* \mathbf{s}_3$, $\mathbf{s}_1 \mapsto^2 \mathbf{s}_4$
- $\mathbf{s}_1 \not\mapsto \mathbf{s}_3$, $\mathbf{s}_1 \not\mapsto^* \mathbf{s}_5$, $\mathbf{s}_2 \not\mapsto^2 \mathbf{s}_4$

Belief Based Unexpected Sequential Patterns

Belief of Sequence

Definition

A belief b of sequence $\mathbf{s}$ is a pair (p, C), where p is a rule between two subsequences $\mathbf{s}_\alpha, \mathbf{s}_\beta \sqsubseteq \mathbf{s}$ and C is a set of constraints:

$$p : \mathbf{s}_\alpha \models \mathbf{s}_\beta$$
$$C : \{\tau, \eta\}$$
$$\tau : n \ \{<, \leq, =, \neq, \geq, >\} \ N \ (N \in \mathbb{N}), n = 0, n = *$$
$$\eta : \{\mathbf{s}_\gamma | \mathbf{s}_\gamma \not\sim \mathbf{s}_\beta\}.$$

denoted as $b = [\mathbf{s}_\alpha; \mathbf{s}_\beta; \mathbf{s}_\gamma; \tau]$. For any two beliefs b_i and b_j, we have:

$$\mathbf{s}_{\alpha i} \sqsubseteq \mathbf{s}_{\alpha j} \Longrightarrow \mathbf{s}_{\gamma i} \not\sqsubseteq \mathbf{s}_{\beta j}.$$

Unexpected Sequence

Definition

Given a belief $b = [\mathbf{s}_\alpha; \mathbf{s}_\beta; \mathbf{s}_\gamma; \tau]$, a sequence $\mathbf{s}$ is *unexpected* if one of the following conditions (*violations*) is satisfied:

1. $\tau : n = *$ and $\mathbf{s}_\alpha \sqsubseteq \mathbf{s}$ (called α *violation*);
2. $\tau : n \neq *$ and $\mathbf{s}_\alpha, \mathbf{s}_\beta \sqsubseteq \mathbf{s}$, $\mathbf{s}_\alpha \mapsto^{\tau'} \mathbf{s}_\beta$, where τ' is incompatible to τ (called β *violation*);
3. $\mathbf{s}_\alpha, \mathbf{s}_\gamma \sqsubseteq \mathbf{s}$ and $\mathbf{s}_\alpha \mapsto^{\tau} \mathbf{s}_\gamma$ (called γ *violation*).

Example of Unexpected Sequences

Example

Given beliefs

$$b_1 = [(A)(B); (C)(D); (E)(F); n = *],$$
$$b_2 = [(A)(B); (C)(D); (E)(F); n = 1].$$

- $\mathbf{s}_1 = \langle (A)(B)(C)(F) \rangle$ is unexpected to b_1 (α violation);
- $\mathbf{s}_2 = \langle (A)(B)(C)(D) \rangle$ is unexpected to b_2 (β violation);
- $\mathbf{s}_3 = \langle (A)(B)(G)(E)(F) \rangle$ is unexpected to b_2 (γ violation);
- $\mathbf{s}_4 = \langle (A)(B)(G)(C)(D) \rangle$ is expected to both b_1 and b_2.

Unexpected Sequential Pattern

Definition

Given an unexpected sequence $\mathbf{s}$ corresponding to belief $b = [\mathbf{s}_\alpha; \mathbf{s}_\beta; \mathbf{s}_\gamma; \tau]$, a segment $\mathbf{s}_u \sqsubseteq \mathbf{s}$ is an *unexpected sequential pattern* if:

1. $\mathbf{s}_\alpha \sqsubseteq^\top_\perp \mathbf{s}_u$ (for α violation); or
2. $\mathbf{s}_\alpha\mathbf{s}_\beta \sqsubseteq^\top_\perp \mathbf{s}_u$ (for β violation); or
3. $\mathbf{s}_\alpha\mathbf{s}_\gamma \sqsubseteq^\top_\perp \mathbf{s}_u$ (for γ violation).

Example of Unexpected Sequential Patterns

Example

Given a belief $b = [(A)(B); (C)(D); (E)(F); n = 2]$.

- $\mathbf{s}_1 = \langle (A)(BD)(DF)(CE)(D) \rangle$ is an unexpected sequence and $\langle (A)(BD)(DF)(CE)(D) \rangle$ is an unexpected sequential pattern;
- $\mathbf{s}_2 = \langle (C)(AC)(BF)(BG)(HF)(CE)(FGH)(E) \rangle$ is an unexpected sequence and $\langle (AC)(BF)(BG)(HF)(CE)(FGH) \rangle$ is an unexpected sequential pattern.

Unexpected Sequential Pattern's Occurrence Problem

Maximal Occurrence Bordered Inclusion

Definition

Given sequences $\mathbf{s} = \langle \mathcal{I}_1 \dots \mathcal{I}_n \rangle$, $\mathbf{s}' = \langle \mathcal{I}'_1 \dots \mathcal{I}'_m \rangle$, $\mathbf{s}'' = \langle \mathcal{I}''_1 \dots \mathcal{I}''_k \rangle$ that $\mathbf{s}'' \sqsubseteq^{\top}_{\perp} \mathbf{s}'$ and $\mathbf{s}' \sqsubseteq \mathbf{s}$. If for any $i > 1$ that $\mathcal{I}'_1 \subseteq \mathcal{I}_i$ there does not exist integer $j < i$ that $\mathcal{I}''_1 \subseteq \mathcal{I}_j$, and if for any $i > 1$ that $\mathcal{I}'_m \subseteq \mathcal{I}_i$ there does not exist integer $j > i$ that $\mathcal{I}''_k \subseteq \mathcal{I}_j$, then $\mathbf{s}'$ is the *maximal occurrence bordered inclusion* sequence of $\mathbf{s}''$ in $\mathbf{s}$.

Example of Maximal Occurrence Bordered Inclusion

Example

$$\mathbf{s}_1 = \langle (A)(B)(C) \rangle$$
$$\mathbf{s}_2 = \langle (B)(A)(AD)(BD)(CD)(C)(B) \rangle$$

The maximal occurrence bordered inclusion sequence of $\mathbf{s}_1$ in $\mathbf{s}_2$ is

$$\langle (A)(AD)(BD)(CD)(C) \rangle .$$

Minimal Occurrence Bordered Inclusion

Definition

Given sequences $\mathbf{s} = \langle \mathcal{I}_1 \dots \mathcal{I}_n \rangle$, $\mathbf{s}' = \langle \mathcal{I}'_1 \dots \mathcal{I}'_m \rangle$, $\mathbf{s}'' = \langle \mathcal{I}''_1 \dots \mathcal{I}''_k \rangle$ that $\mathbf{s}'' \sqsubseteq^{\top}_{\perp} \mathbf{s}'$ and $\mathbf{s}' \sqsubseteq \mathbf{s}$. If any $\mathbf{s}''' \sqsubseteq \mathbf{s}'$ and $\mathbf{s}'' \sqsubseteq^{\top}_{\perp} \mathbf{s}'''$ imply $|\mathbf{s}'''| = |\mathbf{s}''|$, then $\mathbf{s}'$ is the *minimal occurrence bordered inclusion* sequence of $\mathbf{s}''$ in $\mathbf{s}$.

Example of Minimal Occurrence Bordered Inclusion

Example

$$\mathbf{s}_1 = \langle (A)(B)(C) \rangle$$
$$\mathbf{s}_2 = \langle (B)(A)(AD)(BD)(CD)(C)(B) \rangle$$

The minimal occurrence bordered inclusion sequence of $\mathbf{s}_1$ in $\mathbf{s}_2$ is

$$\langle (AD)(BD)(CD) \rangle .$$

Bordered Inclusion and Unexpected Sequential Pattern

Maximal Occurrence Bordered Inclusion

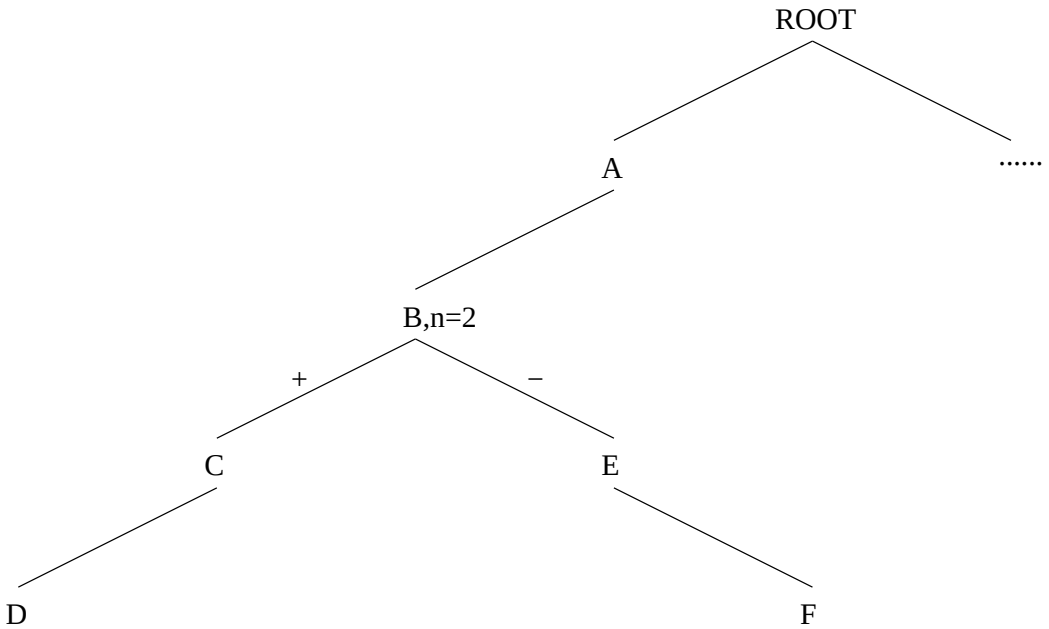
- Maximizes the length of unexpected sequential pattern
- Requires less computational resources

Minimal Occurrence Bordered Inclusion

- Minimizes the length of unexpected sequential pattern
- Requires more computational resources

The USP Approach

Prefix Tree Representation of Belief System



$$b : (p, C)$$
$$p : (A)(B) \models (C)(D)$$
$$C : \{\tau, \eta\}$$
$$\tau : n = 2$$
$$\eta : \{(E)(F)\}$$

USP: Unexpected Sequential Pattern Mining

Algorithm USP^{-}F

Find all unexpected sequential patterns with respect to given belief system.

Algorithm USP^{+}F

Find all unexpected sequential patterns with respect to given belief system, and all frequent sequential patterns with respect to given minimal support.

The USP$^-$F Algorithm

Input

- $\mathcal{B}$: a belief system (candidate sequences) represented as a prefix tree
- $\mathcal{S}$: a database of sequences

Output

- $\mathcal{T}$: Prefix tree containing all unexpected sequential patterns with respect to $\mathcal{B}$

Main Routine of Algorithm USP$^-$F

1 $\mathcal{T} := \emptyset$;
2 **foreach** $\mathbf{s}_b \in \mathcal{B}$ **do**
3 **foreach** $\mathbf{s} \in \mathcal{S}$ **do**
4 $\mathbf{s}_u := \mathit{FindOccurrence}(\mathbf{s}_b, \mathbf{s})$;
5 **if** $\mathbf{s}_u \neq \emptyset$ **then**
6 $\mathit{AppendPrefixTree}(\mathcal{T}, \mathbf{s}_u)$;
7 **end**
8 **end**
9 **end**
10 **return** $\mathcal{T}$;

The USP$^+$F Algorithm

Input

- $\mathcal{B}$: a belief system represented as a prefix tree
- $\mathcal{D}$: a database of sequences
- σ: a minimal support value for frequent sequential patterns

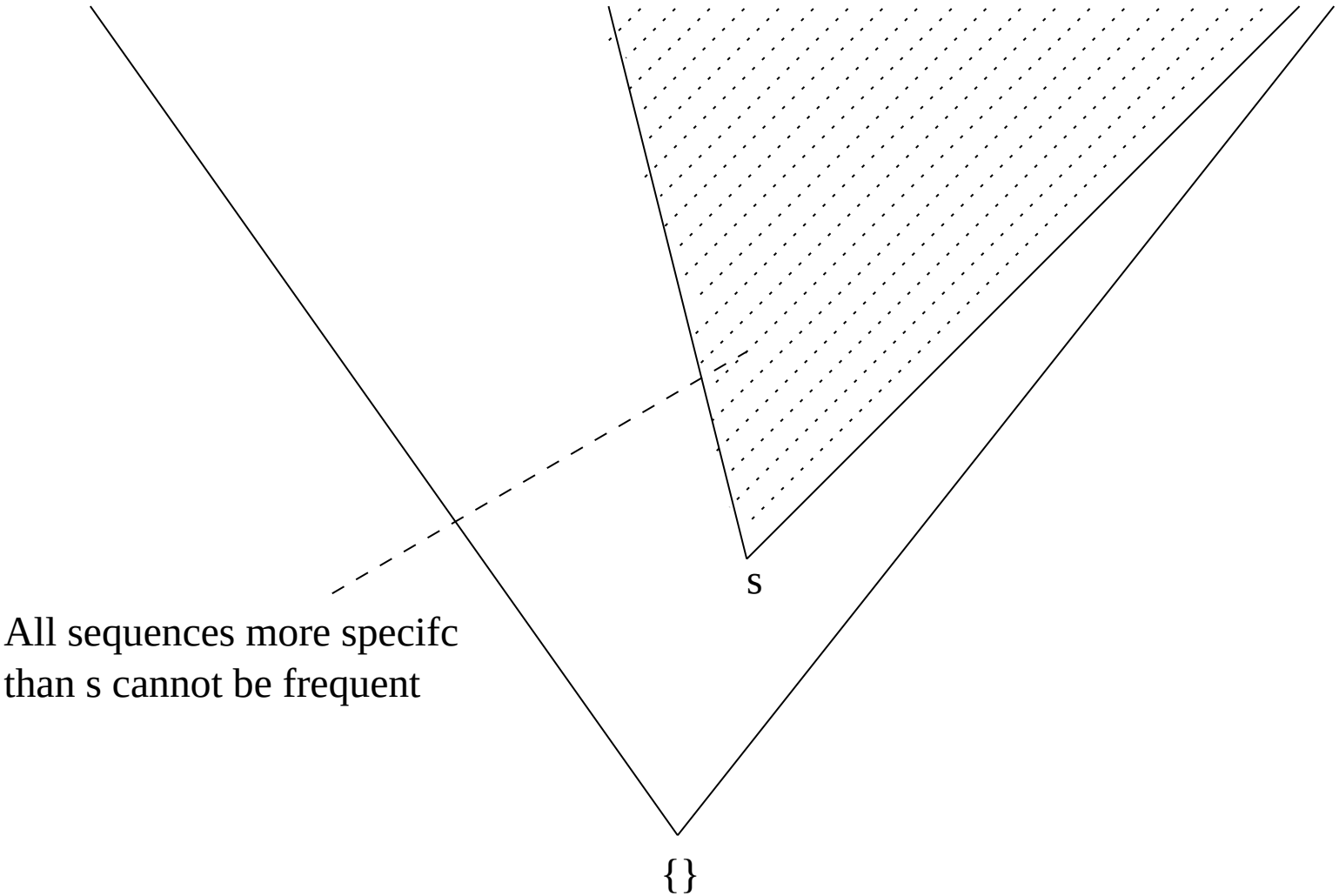
Output

- $\mathcal{T}$: Prefix tree containing of all frequent sequential patterns with respect to σ and all unexpected sequential patterns with respect to $\mathcal{B}$

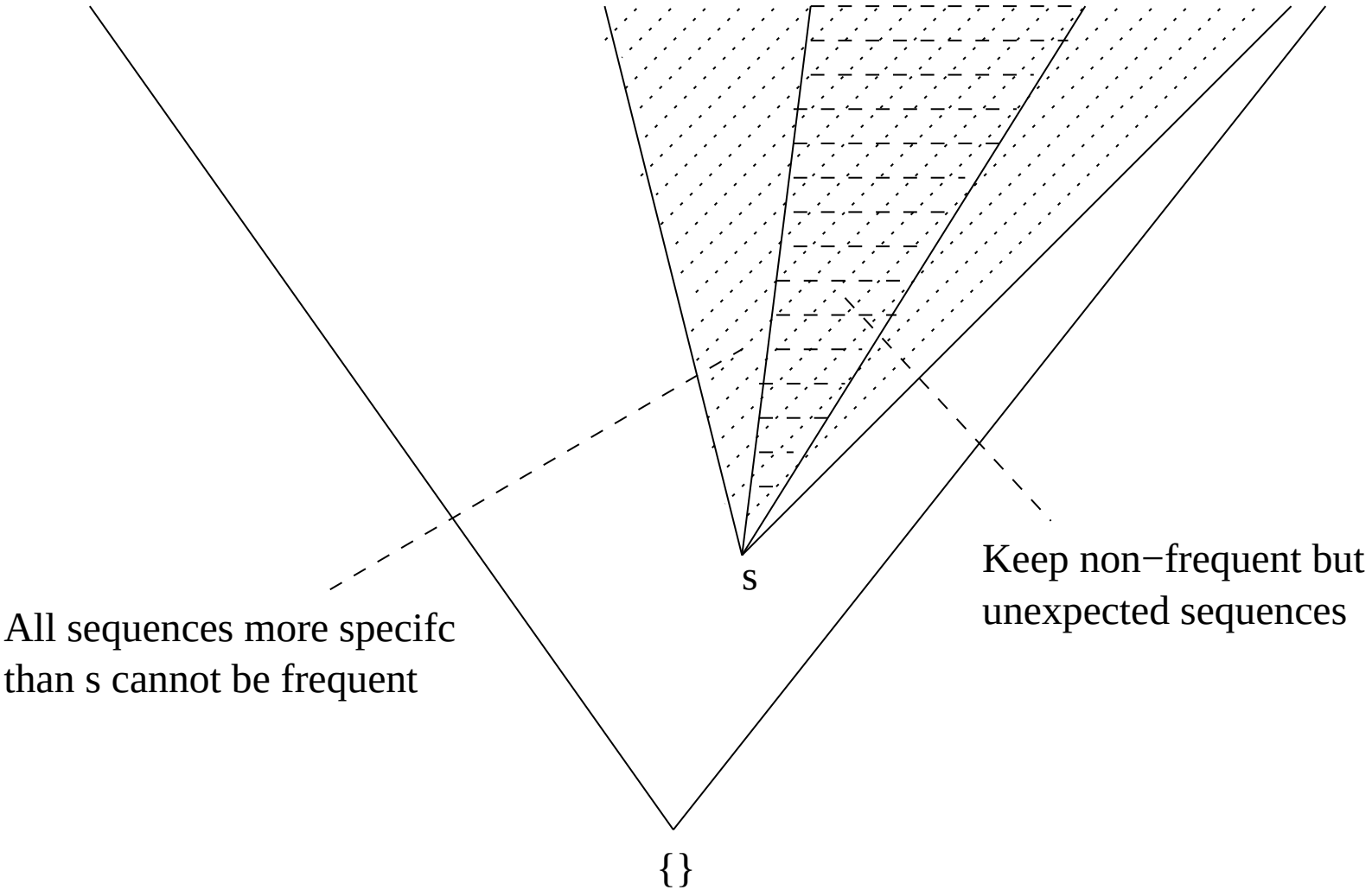
Main Routine of Algorithm USP$^+$F

$\mathcal{T} := \emptyset$;
$k := 1$;
$\mathcal{C}_B := \mathit{AppendBeliefs}(\mathcal{B}, \mathcal{T}, k)$;
$\mathcal{C}_k := \mathit{CountSequence}(\mathcal{D}, \mathcal{T}, \mathcal{C}_B, \sigma, k)$;
while $\mathcal{C}_k \neq \emptyset$ **do**
    $\mathcal{T} := \mathcal{T} \cup \mathcal{C}_k$;
    $\mathcal{C}_B := \mathit{AppendBeliefs}(\mathcal{B}, \mathcal{T}, k+1)$;
    $\mathcal{C}_{k+1} := \mathit{CountSequence}(\mathcal{D}, \mathcal{T}, \mathcal{C}_B, \sigma, k+1)$;
    $k := k + 1$;
end
return $\mathcal{T}$;

Anti-Monotone on Frequent Sequence Spanning



Anti-Monotone on Unexpected Sequence Spanning



Principal of Algorithm USP$^+$F

AppendBeliefs

- Appends first item of each $b \in \mathcal{B}$ to each $\mathbf{i} \in \mathcal{T}$ that does not correspond to any belief.
- Finds unexpected sequential patterns to each $\mathbf{i}_b \in \mathcal{T}$ that corresponds to the first item of any belief.
- Removes all non frequent nodes in current path if $\mathbf{i}_b$ does not occurred in any sequence.

CountSequence

- Counts the frequency of each $\mathbf{i} \in \mathcal{T}$ in each path of current level.
- Finds frequent sequences for next level by the PSP approach.

Outline

Conclusions

1. Sequence Oriented Belief System
2. Unexpected Sequences and Unexpected Sequential Patterns
3. The USP^{-}F algorithm for finding all unexpected sequential patterns
4. The USP^{+}F algorithm for finding all unexpected sequential patterns and all frequent sequential patterns

Perspectives

1. Implementation of the USP approach
2. Experimentation with real world data
3. Proposition of unexpected sequential pattern rules
4. Influence of bordered inclusions on unexpected sequential pattern rules

About Unexpected Sequential Pattern Rules

Anticipation Rule

$$\mathbf{s}_x \Rightarrow \mathbf{s}_u$$

Exception Rule

$$\mathbf{s}_\alpha \Rightarrow \mathbf{s}_y \text{ for } \alpha \text{ violation}$$
$$\mathbf{s}_\alpha \mathbf{s}_y \Rightarrow \mathbf{s}_\beta \text{ for } \beta \text{ violation}$$
$$\mathbf{s}_\alpha \mathbf{s}_y \Rightarrow \mathbf{s}_\gamma \text{ for } \gamma \text{ violation}$$

Influence Rule

$$\mathbf{s}_u \Rightarrow \mathbf{s}_z$$

Thank you!