Explicit and Implicit Pattern Relation Analysis for Discovering Actionable Negative Sequences

Wei Wang and Longbing Cao, Senior Member, IEEE

Abstract—Real-life events, behaviors, and interactions produce sequential data. An important but rarely explored problem is to analyze those nonoccurring (also called negative) yet important sequences, forming negative sequence analysis (NSA). A typical NSA area is to discover negative sequential patterns (NSPs) consisting of important nonoccurring and occurring elements and patterns. The limited existing work on NSP mining relies on frequentist and downward closure property-based pattern selection, producing large and highly redundant NSPs, nonactionable for business decision-making. This work makes the first attempt for actionable NSP discovery. It builds an NSP graph representation, quantifies both explicit occurrence and implicit nonoccurrence-based element and pattern relations, and then discovers significant, diverse, and informative NSPs in the NSP graph to represent the entire NSP set for discovering actionable NSPs. A DPP-based NSP representation and actionable NSP discovery method, EINSP, introduces novel and significant contributions to NSA and sequence analysis: 1) it represents NSPs by a determinantal point process (DPP)-based graph; 2) it quantifies actionable NSPs in terms of their statistical significance, diversity, and strength of explicit/implicit element/pattern relations; and 3) it models and measures both explicit and implicit element/pattern relations in the DPP-based NSP graph to represent direct and indirect couplings between NSP items, elements, and patterns. We substantially analyze the effectiveness of EINSP in terms of various theoretical and empirical aspects, including complexity, item/pattern coverage, pattern size and diversity, implicit pattern relation strength, and data factors.

Index Terms—Actionable pattern discovery, determinantal point process (DPP), explicit relation, implicit relation, negative sequence analysis (NSA), negative sequential pattern (NSP), nonoccurring behavior analysis (NBA), pattern mining, pattern relation analysis.

I. INTRODUCTION

A. Nonoccurring/Negative Sequence Analysis

NEGATIVE sequence analysis (NSA) [1] is a typical method for nonoccurring behavior analytics [2]. NSA aims to discover interesting negative sequences consisting of negative (nonoccurring) and positive (occurring) elements in a sequence. A typical research area in NSA is to discover negative sequential patterns (NSPs) [3], [4], [5] that consist of frequent negative sequences. Each element may consist of one to multiple items, e.g., behaviors. An NSP consists of several negative elements, which may also have positive ones as well. Here, negative denoted by the symbol “−” means nonoccurring, e.g., the absence (nonoccurrence) of a behavior (an item) or behavior sequence (forming an element or itemset) that is important yet undeclared or missing.

Examples of negative items are hiding external income (thus undeclared in transactions so as to obtain government low-income allowance) and missing an important appointment (such as a medical treatment that may incur serious health issues). For example, in COVID-19, an NSP could be $p = \langle$attending-party$, \langle$wearing-mask$, \langle$social-distancing$\rangle\rangle \rightarrow\langle$Infected$, \langle$attending-party$, \langle$wearing-mask$, \langle$social-distancing$\rangle\rangle \rightarrow\langle$Infected$, \langle$attending-party$, \langle$wearing-mask$, \langle$social-distancing$\rangle\rangle = \langle$wearing-mask$, \langle$social-distancing$\rangle$ as an negative element, and (attending-party, $\langle$wearing-mask$, \langle$social-distancing$\rangle\rangle$) as a negative sequence. We use “,” to separate elements, where an element consisting of more than one item is a compound element, while an element with only one item is a single item element. Pattern $p$ is an impact-targeted NSP with an impact label Infected [6]. Note that, often, impact labels are not necessarily included in NSP and positive patterns. Pattern $p$ indicates that one who attends parties without wearing masks and maintaining social distancing has a high probability of being infected by COVID-19. This NSP reiterates the importance of wearing masks and social distancing to avoid COVID-19 infection, where infection likely occurs in the absence of two strongly encouraged precautionary behaviors wearing-mask and social-distancing.

NSPs are usually more informative and useful than positive sequential patterns (PSPs) (e.g., [7], [8], and [9]) as they can disclose nonoccurring but important behaviors [2]. NSA mining can disclose the unique NSP value of modeling sequential nonoccurring entities, behaviors, and events, including understanding negative sequences [5], nonoccurring yet important behaviors [2], and complex behavioral relationships [6], [10], [11]. They have been applied to business problems, such as fraudulent health insurance claim detection [3], missing medical treatment detection [12], [13], [14], debt detection in taxation and social welfare services [6], [15], [16], [17], overservice detection [6], [16], and factors associated with poor academic performance [18].

B. Gap Analysis and Target Problem

However, both NSA and NSP discovery involve significant theoretical and practical challenges. Examples are modeling negative elements with diverse formats and combinations; negative patterns mixed with negative items and elements...
in sophisticated combinations; various negative containment scenarios [3]; and different couplings between negative and positive items, elements, and patterns [19]. Addressing these learning issues not only requires new theories and tools but also results in high computational cost, large frequent but overlapped findings, and missing significant yet infrequent behaviors, making existing NSA and NSP nonactionable [3], [13], [19], [20]. These challenges cannot be directly addressed by existing PSP mining methods [1], [3], general sequence analysis methods, and deep neural network-based sequential modeling [21]. They only model the occurring items in a sequence but do not handle sophisticated structures and relations in the NSP format, combination, or containment. Very limited theoretical and algorithmic work is available to effectively and efficiently address these learning challenges.

To address the aforementioned issues, we propose to discover actionable NSPs that: 1) are representative, highly probable, and diverse to represent the whole original NSP collection; 2) filter highly similar and redundant patterns for low computational complexity and high efficiency [3]; and 3) are discriminative and informative for suggesting decision-making actions [19]. We call this actionable NSP discovery, motivating this work. Actionable NSP discovery involves several essential but difficult issues: 1) analyzing the relations between NSP elements and patterns, i.e., pattern relation analysis [19] to understand how NSP patterns interact with each other, which has rarely been explored; 2) efficiently representing the explicit (e.g., occurrence-based) and implicit (e.g., hidden and indirect) couplings between positive and negative items, elements, and their combinations [1], unexplored in NSA; 3) efficiently handling various sequential combinatorial issues related to item and element combinations, formatting, and containment, very challenging to achieve; and 4) quantifying the optimal selection criteria on each subset candidate in the whole NSP collection (with an emphasis on quality rather than frequency). No existing research on NSA, behavior computing, sequence analysis, and deep sequential modeling addresses these challenges. SAPNSP [20] appears to be the only one somehow relevant to representative NSP mining. It follows traditional frequentist by applying the lift-based contribution measure proposed in [16] to evaluate the contribution of each pattern and select those highly significant ones as the top-k recommendations. It, however, completely ignores the complex couplings between elements and between patterns [19], [22], [23] and the statistical significance of such pattern relations in selecting NSPs, thus producing many redundant and similar patterns.

C. Main Design and Contributions

In actionable NSP discovery, on the one hand, since elements in the NSP cohort often form an imbalanced distribution, patterns that satisfy the frequentist selection criteria are often similar and indicate less informative knowledge. The discovered NSPs are associated with low diversity, low coverage, and often a heavy-headed distribution of pattern subsets, which miss those infrequent but important items and, thus, are nonactionable [24], [25]. The frequentist-based NSA methods and pattern selection criteria widely used in PSP mining and sequence analysis filter patterns with rarely observed items due to their sparsity and the biased selection criteria. In reality, some of the long-tailed items may stand for rare but vital behaviors and are critical in specific situations [2], such as suspicious health claims in fraud detection, fault maintenance in system diagnosis, and missing treatments in medical services. On the other hand, the NSPs selected by frequency-based measures and per the downward closure property are often short in size and duplicate, making the discovered pattern subsets less capable of disclosing long-range behaviors and informing decision-making actions. This indicates the importance of measuring NSP actionability, such as quality, frequency, probability, diversity, and informativeness [19] in actionable NSP discovery to serve multiple purposes. One is to select high-quality patterns that carry important multiaspect information about the representative characteristics in the data. The other is to discover diverse-sized patterns as a group that ensures nonrepetitive but informative subset representatives. Intuitively, pattern diversity implies a repulsive interaction and negative dependence between NSPs so that similar ones less likely co-occur [26], [27]. In the end, those patterns of low frequency but with more informative knowledge are retained and informative.

Accordingly, we apply determinantal point processes (DPPs) to represent NSPs as a DPP-based NSP graph, model the explicit (directly linked) and implicit (indirectly linked) pattern relations [19], and then discover actionable NSP subsets. DPP as an efficient probabilistic model captures negative correlations for subset selection by keeping the diversity in the subset [28], [29], [30], [31], [32]. However, DPP cannot discover actionable NSPs. Building on our work on implicit coupling learning [22] and its application in inferring indirect rules [33], this work introduces non-co-occurrence-based implicit relations between NSP itemsets in the DPP-based NSP graph and captures the implicit couplings between two NSP's (or NSP itemsets, e.g., paths \( p_1 \) and \( p_2 \) conditioned on other NSP elements or itemsets (e.g., path \( p_3 \)). This method is even more effective in discovering implicitly coupled elements and patterns, being more informative, and discovering even unexpected hidden knowledge [10], [19]. Accordingly, an explicit and implicit element/pattern relation-based actionable NSP discovery method EINSP conducts the DPP-based actionable NSP discovery.

1) This is the first work on designing a DPP-based graph representation of NSPs and discovering high-quality NSPs that are representative and of high probability and high diversity in the NSP graph. The problem of representative NSP discovery is converted to a probabilistic subset selection problem in a DPP graph, which takes advantage of the probabilistic DPP theories and graph representation strength in subset selection with diversity. We derive the DPP-based theory for representative NSP discovery and design EINSP to select a subset of informative and diverse NSPs using DPP-based subset selection.
2) EINSP captures rich element interactions and pattern relations in NSPs, which are rarely explored in existing NSA and sequence analysis. The co-occurring and
nonoccurring element and pattern relations in NSPs are modeled in terms of the direct and indirect DPP-based node/edge dependencies, which characterizes the probability and diversity of each NSP pattern in the NSP collection in terms of both explicit element/pattern co-occurrences and implicit nonocurrences conditional on third parties. EINSP integrates both explicit and implicit element/pattern relations in the DPP-based NSP graph and effectively samples those highly explicitly and implicitly coupled NSPs as a representative subset of high-probability and diverse NSP patterns.

We verify EINSP on six real datasets and 17 synthetic datasets against baseline methods in terms of sequence and item coverage, average pattern size, average implicit relation strength (IRS), and sensitivity to various data factors, including scalability. The substantial empirical analysis demonstrates that EINSP achieves significant performance in discovering representative NSPs. It opens new opportunities for efficient NSA through DPP-based representation and learning.

II. BACKGROUND AND RELATED WORK
A. NSA, NSP Mining, and Nonoccurring Behavior Analysis

Sequence analysis\(^1\) typically discovers patternable combinations of items/elements in sequences, e.g., frequent sequential patterns [8], [23], which is also termed (positive) sequence analysis (PSA) [7], [34], [35], [36], [37], [38], in particular, PSP mining [8], [9]. PSA and PSP minings identify frequently co-occurring itemsets. In contrast, NSA and NSP minings [1], [3] discover nonoccurring but interesting items/elements and itemsets. Typically, PSA and NSA focus on selecting frequent patterns.

While PSA and, in particular, PSP mining have been intensively explored, much less effort has been dedicated to developing basic theories and efficient computational tools for NSA and NSP minings in its over-a-decade history [1]. This is due to the intricate characteristics and challenges of negative items, elements, and sequences [3] and the fact that the approaches and tools developed for PSA and PSP minings cannot be directly transferred or lightly adapted to NSA and NSP minings since they hold highly different problem settings and complexities. Existing work on NSA and NSP minings involves various aspects of constraints to restrict the settings of negative candidates and NSPs, and reduce or control the high combinatorial challenge and computational complexity [1], [3]. Examples are element size constraints, element format constraints, pattern structure constraints, and containment constraints. Note that even the same constraint settings [1], [39] could incur different challenges and solutions in PSP and NSP mining. Furthermore, NSA only forms a small set of the even broader and more challenging problem—nonoccurring behavior analysis (NBA) [2]. NBA involves many rarely considered problems, settings, and opportunities of behaviors that are important but have not yet happened (nonoccurred) or observed (unobserved or hidden).

Major approaches for NSA and NSP minings can be categorized into two: 1) the frequentist-based NSA that develops frequency-based statistics, including support and confidence of negative candidates and NSPs and for their selection [1], [3], [12], [40] and 2) the set theory-based NSA and its landmark algorithm e-NSP proposed in [3] by converting NSA to PSP and then conducting PSP-based NSA. The former approach transfers the theories of PSA and PSP minings to NSA and NSP minings but cannot handle the fundamental challenges of NSA due to their intrinsic differences. As a result, very limited progress has been made so far. The latter approach opens a fundamentally new direction to address the high combinatorial challenge and computational complexity in NSA, leading to various recent follow-ups [41], [42], [43], [44], [45]. However, their methodological dependence on frequentist statistics incorporates various constraints (e.g., FRI in [45]) rules out infrequent items and elements, and none of them involves relation modeling of elements and patterns as in our work.

In contrast, this article opens a new direction for NSA postprocessing. It avoids the frequentist-oriented combinatorial challenge and computational complexity by converting the NSA problem to subgraph selection after designing a DPP-based graph representation of NSPs. This work is, thus, highly fundamental and promising for addressing some of the critical challenges in NSA.

B. Actionable NSP Discovery

NSA and NSP minings still face some critical theoretical and computational challenges. One is to develop general representations of negative sequences and their various settings (e.g., on format, type, and containment) and constraints on the combinations of items, elements, and sequences. Another is the evaluation and selection of quality NSPs (here, “quality” may refer to aspects of significance, representativeness, novelty, coverage, and actionability). The last example is to analyze explicit and implicit coupling relationships within sequences and between sequences [7], [19], [22], [23], e.g., pair patterns and cluster patterns [6], [19].

The challenge addressed in this work is on discovering actionable NSPs [19], [20] that are significant, diverse, and informative for suggesting decision-making actions. This is essential and critical to not only discover more actionable NSPs to enhance NSA applicability, such as for next-best medical treatments, but also to address some fundamental technical issues, such as analyzing sophisticated couplings in NSP elements and sequences. Compared with the high degree of attention paid to general NSA discovery (e.g., [3], [4], [12], and [13]), very limited research has been conducted on selecting representative NSPs that are not only statistically significant but also more actionable (e.g., inducing more diversified and novel elements) to decision-making.

However, to the best of our knowledge, SAPNSP [20] is the only method for NSP subset selection, i.e., extracting top-k NSPs from those NSP collections discovered by other NSP miners. SAPNSP cannot discover actionable NSPs due to three main design shortages: 1) it only applies a single measure contribution originally proposed in our early work in [16] to selecting patterns of high frequency and high correlation between its prefix and the last element, producing a highly repetitive resultant subset; 2) the contribution

\(^1\)Note that, here, we do not involve the general sequence analysis, such as deep sequential modeling and other topics, including time series analysis.
measures are based on lift and the downward closure property, which, thus, overestimates the quality of short-size patterns but filters long-size patterns with highly frequent and informative elements; and 3) SAPNSP does not consider implicit nonoccurrence couplings between elements and between NSPs and, thus, filters these lowly frequent but informative patterns. Our approach suggests a comprehensive solution without the aforementioned limitations in discovering actionable NSP discovery. In comparison to contribution-based filtering, EINSP calculates both co-occurrence- and non-co-occurrence-based dependencies between directly and indirectly linked itemsets in the whole NSP cohort. Thus, they capture much richer interactions than contribution between NSP elements and itemsets, and are not restricted to the downward closure property-based element overlapping as contribution. In addition, although SAPNSP addresses the interestingness of mining NSPs, its application of contribution-based filtering is too simple without an appropriate measure of NSP “actionability” [19] and any empirical evaluation of the effectiveness as in our work (see more details in Section V for a comprehensive evaluation of EINSP effectiveness).

### C. DPP and DPP-Based Subset Selection

DPPs have shown great strength in subset selection by modeling the probability over all subsets in terms of their quality and diversity. DPP has shown promise for video and documentation summarization [21], [24], [46], [47], information retrieval [48], recommender systems [49], sequence classification [36], and image processing [34], [50]. However, DPP has not been used for actionable NSP discovery. On the one hand, NSPs are embedded with sequential structures, which cannot be modeled by the point-based DPP methods. So far, structured DPP (SDPP) [51] is the only DPP-based method for distributions over sets of structures and was applied to documentation summarization [48]. However, it only models the quality and diversity of structures in terms of a single relation but fails to handle multiple complex relations between entities, which may cause unacceptable information loss [34]. On the other hand, the relations between NSPs are much more complicated than the dependence modeled by the existing DPP-based methods. The method in [48] incorporates cosine similarity into SDPP to model the quality of each entity and quantify the diversity between two entities (an entity can be a structure or a component within the structure), which only reveals the co-occurrence relations. In addition, k-DPP only selects a significant and diverse subset with cardinality k [50]. MDPP refined on k-DPP selects an optimal fixed-size subset by modeling multiple relations among the entities in the collection [34]. However, neither k-DPP nor MDPP is applicable for selecting structural patterns, such as NSPs.

Our work represents the first attempt at engaging DPP for NSA mining for NSA quality enhancement and representative subset selection. No research has been reported on effectively representing NSA in terms of DPP. Our proposed EINSP represents the highly probable, diverse, and coupled NSPs by DPP, jointly modeling co-occurrences and nonoccurrence-based explicit and implicit itemset/pattern relations [19] that have rarely been explored but are very important to PSA and NSA research, and selecting representative NSP subsets as the NSA output.

### III. EINSP METHOD

#### A. Problem Statement and System Framework

Let us illustrate the aforementioned motivation of actionable NSP discovery with a more general healthcare analytical case. Assume that \( p = (a, b, c, X) \) is an NSP, where \( a, b, \) and \( c \) stand for the codes of the medical services undertaken on a patient, and \( X \) is the final disease status of the treatment. \( p \) indicates that patients who undertake medical services \( a \) and \( c \) without treatment \( b \) have a high probability of having disease status \( X \), which shows the impact of the absence of \( b \) on status \( X \). Pattern \( p \) may correspond to a special treatment of a serious but low-chance disease, such as rare cancer, where treatments \( a, b, \) and \( c \) are less likely to co-occur frequently. Such patterns may not be selected by existing NSA methods due to their relatively low frequency even though \( p \) may be extremely useful for informing the disease treatment. This example also shows the importance of involving the negative items/elements and their implicit relations with positive items/elements, both of which are usually ignored in PSP and general sequence modeling; the findings are thus more informative by referring to the nonoccurring items and infrequent patterns. However, none of the existing DPP-based subset selection methods can model such pattern quality and diversity-related settings, and can jointly model positive and negative element/pattern relations.

To discover actionable NSPs, Fig. 1 illustrates the working process of the proposed DPP-based actionable NSP discovery EINSP. EINSP consists of four components: NSP graph construction, explicit relation modeling, implicit relation modeling, and overall relation-based NSP subgraph selection. First, the NSP graph construction step transforms a collection of NSPs discovered by an NSP miner (e.g., Negative-GSP [12]) into a directed DPP-based graph. Second, the explicit relation modeling step models the explicit co-occurring relations between NSPs in the DPP graph and computes the probability of selecting an NSP subset with such explicit relations. Third, the implicit relation modeling step models the implicit nonco-occurring relations between NSPs with regard to other NSPs in the DPP graph and computes the probability of selecting an NSP subset with such implicit relations. Finally, the overall relation-based selection computes the overall probability of selecting an NSP subset by integrating both explicit relation-based probability and implicit relation-based probability. A representative NSP subset is selected from the DPP-based NSP graph per the overall probability and selection criteria. EINSP selects k-size informative and diverse subsets from the original NSP cohort in the DPP graph.

Accordingly, let us assume \( \mathcal{Y} \) be the entire NSP pattern set; \( Y_i \in \mathcal{Y} \) refers to the \( i \)th NSP, and \( y'_j \) refers to the \( j \)th element in pattern \( Y_i \). Correspondingly, their notations in the converted DPP-based graph are given as follows: \( \mathcal{G} \) refers to the NSP-converted DPP graph corresponding to the pattern set \( \mathcal{Y} \); \( Y_i \) refers to the path in the DPP graph corresponding to NSP \( Y_i \), which represents a vectorized representation of the NSP in \( \mathcal{G} \); and \( y'_j \) refers to the node in graph \( \mathcal{G} \) corresponding to the NSP element \( y'_j \), i.e., an element of vector \( Y_i \).
Furthermore, assume that we have a collection of NSPs \( \mathcal{Y} = \{Y_1, Y_2, \ldots, Y_N\} \) discovered by any existing NSP miner as the input, where \( Y_i = (y^1_i, y^2_i, \ldots, y^m_i) \) is an NSP and \( y^j_i \in Y_i \) is an NSP element in the NSP. EINSP first transforms NSPs \( \mathcal{Y} \) into a directed DDP-based graph \( \mathcal{G} \), which is a powerful and compressed representation of the NSPs. Second, EINSP models the explicit relations between NSPs in the DPP graph and computes the probability \( P_e^k(Y) \) of selecting an NSP subset \( Y \in \mathcal{Y} \) in terms of co-occurrences between NSPs in each \( k \)-subgraph in the DPP graph. Then, EINSP further represents the implicit relations between NSPs in the DPP graph and computes the probability \( P_i^k(Y) \) of selecting the NSP subset \( Y \) in terms of their nonoccurrences with other NSPs. Furthermore, EINSP calculates the overall probability of the NSP subset \( Y \) by integrating both \( P_e^k(Y) \) and \( P_i^k(Y) \). Finally, EINSP selects the representative \( k \)-size NSP subsets as the actionable NSPs, which correspond to the highly probable, diverse, and explicitly and implicitly coupled NSP subgraphs in the DPP-based NSP graph per selection criteria over the overall probability \( P^k(Y) \). In the following, we introduce each of these modules in detail.

**B. NSP Graph Construction**

In general, the NSP collection \( \mathcal{Y} \) is always on a large scale and many patterns share identical subsequences (for simplicity and consistency with the terms in NSP, subsequences are used interchangeably with *itemsets*) due to the element and itemset co-occurrences [3]. Hence, the NSP graph construction module converts \( \mathcal{Y} \) to a directed graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \), where the direction refers to the sequential order between elements in NSPs. In \( \mathcal{G} \), each node \( y \in \mathcal{V} \) corresponds to an element (consisting of all items in the element) of an NSP pattern, and each edge \( e \in \mathcal{E} \) stands for a directed linkage from one element to another in an NSP, which reflects the sequential co-occurrences between the elements in the NSP. In this way, each NSP \( Y_i \in \mathcal{Y} \) is transformed into a directed path of \( \mathcal{G} \).

In \( \mathcal{G} \), we define\(^2\) the *explicit element quality*. It quantifies the quality (significance) of each NSP element (i.e., a node significance in the graph) in a graph \( \mathcal{G} \). We further define the *explicit element pair quality* for each element pair in an NSP (i.e., an edge in \( \mathcal{G} \)) in terms of the explicit co-occurrence relations between NSP elements.

**Definition 1 (Explicit Element Quality):** Given a path \( Y_i = (y^1_i, y^2_i, \ldots, y^m_i) \) in \( \mathcal{G} \), which corresponds to an NSP pattern \( Y_i \) in the pattern set \( \mathcal{Y} \), the *explicit element quality* \( q_e(y^j_i) \) measures the significance of pattern element \( y^j_i \) in \( \mathcal{G} \), and the *explicit element pair quality* \( q_e(y^j_i, y^{j+1}_i) \) measures the co-occurrence-based significance of the pattern element pair \( (y^j_i, y^{j+1}_i) \) in \( \mathcal{G} \).

In addition, we define the *explicit element diversity feature vector* for each NSP element in \( \mathcal{G} \).

**Definition 2 (Explicit Element Diversity Feature Vector):** Given a node \( y^j_i \in Y_i \) in \( \mathcal{G} \), which corresponds to an element \( y^j_i \) of NSP \( Y_i \), its *explicit element diversity feature vector* \( \phi_e(Y^j_i|E) \) measures the explicit co-occurrence diversity between NSP element \( y^j_i \) and other co-occurring NSP elements in \( \mathcal{G} \), where \( E \) is the set of potential elements of NSPs and \( |E| \) is the size of all elements in \( E \).

Finally, we define the *implicit pattern quality* and the *implicit pattern diversity feature vector* for each NSP in \( \mathcal{G} \).

**Definition 3 (Implicit Pattern Quality):** Given a path \( Y_i \) in graph \( \mathcal{G} \), which corresponds to an NSP \( Y_i \) in the pattern set \( \mathcal{Y} \), the *implicit pattern quality* \( q_i(Y_i) \) measures the quality of the path in \( \mathcal{G} \) in terms of implicit relations between paths, i.e., the nonoccurrence level between \( Y_i \) and other NSPs in the DPP graph.

**Definition 4 (Implicit Pattern Diversity Feature Vector):** Given a path \( Y_i \) in graph \( \mathcal{G} \) corresponding to an NSP \( Y_i \) in the pattern set \( \mathcal{Y} \), the *implicit pattern diversity feature vector* \( \phi_i(Y_i) \) measures the diversity of the path in \( \mathcal{G} \) in terms of the nonoccurrence relations between \( Y_i \) and other paths, i.e., the nonoccurrence level between pattern \( Y_i \) and other NSPs in the DPP graph.

The above basic concepts form the base for the following formalization of DPP-based NSP representation and learning.

**C. Explicit Relation Modeling**

Since each pattern \( Y_i \in \mathcal{Y} \) represents an ordered structure of NSP elements, the occurrence probability (also called *explicit probability*) of an NSP subset \( Y \) can be computed by a fixed \( k \)-size structured determinatal point process
The representation $C^c$ can be calculated by a second-order message passing algorithm [29], [53].

Once $C^c$ is computed, it can further be eigen-decomposed in the form of $C^c = \sum_{n} \lambda_n^c v_n^c (v_n^c)^T$ in time $O(|E|^3)$. We then obtain the eigenvalue/eigenvector pairs $(\lambda_n^c, v_n^c)$ of representation $C^c$, where $N_c$ is the number of eigenvalue/eigenvector pairs. As proved in [28], the nonzero eigenvalues of $C^c$ and $L^c$ are identical. If $v_n$ is the nth eigenvector of $C^c$, then $B^c v_n$ is the nth eigenvector of $L^c$, which shares the same eigenvalue $\lambda_n^c$. That is, $(\lambda_n^c, B^c v_n^c)$ are the corresponding pairs of kernel $L^c$, making $L^c = \sum_{n} \lambda_n^c (B^c v_n^c) (B^c v_n^c)^T$.

To efficiently select a subset of $k$ patterns, inspired by the mechanism of k-DPP [50], we formalize $P^c_k(Y)$ as follows:

$$P^c_k(Y) = \frac{1}{C^c \leq k} \prod_{n=1}^{N_c} P^c(Y) \prod_{n=1}^{N_c} \lambda_n^c.$$

Here, $J$ is the index subset of $(\lambda_n^c, B^c v_n^c)_{n=1}^{N_c}$, $V^c_j$ stands for the eigenvector subset indexed by $J$, i.e., $V^c_j \equiv \{B^c v_n^c_{n=1}^{N_c} \}_{n=J}$. In addition, $e^c_{k, N_c} = \sum_{J} (e^c_{k, N_c})_{j=1}^{N_c} \prod_{n=1}^{N_c} \lambda_n^c$ is the $k$th

(k-SDPP) model [51], which captures the co-occurrence-based relationships between NSP elements and their formation into NSP patterns. The k-SDPP model captures the distribution of pattern subset $Y \subseteq \mathcal{Y}$ conditional on the event that subset $Y$ has cardinality $k$ as follows:

$$P^k(Y) = \frac{\det(L'_Y)}{\sum_{|Y'|=k} \det(L'_Y')} \quad (1)$$

Here, $L'$ is a positive semidefinite kernel, $L'_Y$ represents the restriction on $L'$ to entries indexed by NSP subset $Y$ [50], i.e., $L'_Y \equiv \{L'_y|y_j \in Y\}$, and $\det(L'_Y)$ denotes the determinant of $L'_Y$.

Following the design in [28], kernel $L'$ can be rewritten as a Gram matrix: $L' = B' \times B'$. $B' \in \mathbb{R}^{E \times E}$ is the matrix where each column is a feature vector [28], describing the corresponding pattern in the NSP collection $\mathcal{Y}$. Column $B'_y$ can be factorized as the production of an explicit pattern quality score $q_e(Y)$ and a normalized explicit pattern diversity feature vector $\phi_e(Y)$, $\|\phi_e(Y)\| = 1$. Here, $q_e(Y) \in \mathbb{R}^E$ can be viewed as a nonnegative measure of the explicit significance of pattern $Y$, and $\phi_e(Y) \equiv \sum_{Y \subseteq \mathcal{Y}} q_e(Y)$ for two elements $y_j, y_j'$ and their element pair $y_j, y_j'$.

Accordingly, the entries of kernel $L'$ can be further decomposed as follows:

$$L'_Y = q_e(Y) \phi_e(Y)^T \phi_e(Y) q_e(Y). \quad (2)$$

Equation (2) gives rise to a distribution [24], which places a higher weight on the NSP subsets that are composed of higher quality and more diverse NSP patterns.

To efficiently define a DPP over the sequential NSP patterns, the explicit pattern quality score $q_e(Y)$ is multiplicatively decomposed as follows. We use a log-linear model to decompose it, which depends on the explicit quality of each element $q_e(y_j)$ and its element pair $q_e(y_j, y_j')$ for two elements $y_j$ and $y_j'$.

$$q_e(Y) = \exp\left(\sum_{j=1}^{N} q_e(y_j) + \sum_{j=1}^{N-1} q_e(y_j, y_j')\right) = \prod_{j=1}^{N} \exp(q_e(y_j)) \times \prod_{j=1}^{N-1} \exp(q_e(y_j, y_j')). \quad (3)$$

As NSP discovery is built on frequentist statistics, we specify the explicit quality of a pattern feature $d$ (corresponding to the element and element pair in the feature set) in terms of element and element pair frequencies in NSPs, i.e., $q_e(d) = \frac{(|D \land (d \subseteq s_d)|/|D|)}{s_d}$, where $D$ denotes the pattern set, and $s_d$ is an NSP sequence from dataset $D$. Similarly, the normalized explicit pattern diversity feature vector $\phi_e(Y)$ of an NSP pattern $Y$ is decomposed additively over the elements $y_j$ of $Y$:

$$\phi_e(Y) = \text{Norm}\left(\sum_{j=1}^{N} \phi_e(y_j)\right). \quad (4)$$

Here, $\text{Norm}(\cdot)$ guarantees that, for any pair of NSP patterns $Y$ and $Y_k$, $\phi_e(Y_k)^T \phi_e(Y_k) \equiv [1, 1]$ is a signed measure of the similarity between these patterns. The diversity feature $d_k^c(y_j) = \phi_e(y_j)$ is the $k$th component of vector $\phi_e(Y)$. The $k$th component is identified by the explicit normalized elementwise mutual information (eNEMI) between NSP element $y_j$ and other NSP elements $E_k \in E$. eNEMI is adapted from the normalized pointwise mutual information (NPMI) [52].

Definition 5 (eNEMI): The eNEMI $\text{NEMI}(y_j^c, E_k)$ measures the ability to capture both linear and nonlinear dependencies between element $y_j^c$ and other NSP elements $E_k \in E$, which is defined as follows:

$$d_k^c(y_j) \equiv \text{eNEMI}(y_j^c, E_k) = \frac{h(y_j^c) + h(E_k) - h(y_j^c, E_k)}{h(y_j^c)} \quad (5)$$

where $h(\cdot) = -\log(p(\cdot))$, and $p(y_j^c)$ and $p(E_k)$ are the marginal probabilities of NSP elements $y_j^c$ and $E_k$ in graph $G$, $p(y_j^c, E_k)$ is the joint probability. Note that $d_k^c(y_j^c) \in [-1, 1]$, and some orientation values [52] are given as follows: 1) when NSP elements $y_j^c$ and $E_k$ occur separately but never co-occur, they hold a negative dependence corresponding to non-co-occurrents between NSP elements, $d_k^c(y_j^c) = -1$; 2) when they are distributed under independence, $d_k^c(y_j^c) = 0$ as the numerator is 0; and 3) when they completely co-occur, they hold positive dependence corresponding to the full co-occurrences between NSP elements, $d_k^c(y_j^c) = +1$. Hence, $\phi_e(y_j^c)$ serves as an element dependence measure of $y_j^c$ w.r.t. all other elements.

Discovering NSP involves a huge search space and a high computational cost. The size of the NSP collection $\mathcal{Y}$ is always enormous in [3], [5], [12], which makes kernel $L'$ too large and the inference on $L'$ intractable. Accordingly, a dual representation of $L'$, i.e., $C' = B' \times B'^T$, is constructed to represent the properties carried by kernel $L'$ to enable efficient inference [28]. Note that $C' \in \mathbb{R}^{E \times E}$ is a symmetric and positive semidefinite matrix, where, typically, $|E| \ll |\mathcal{Y}|$. Hence, $C'$ is always much smaller in scale and less sensitive to a threshold than kernel $L'$.
elementary symmetric polynomials on eigenvalues, which is equivalent to the normalization constant in (1) and can be computed by a recursive algorithm. Finally, \( P^{V_j} \) denotes an elementary DPP with marginal kernel \( K^{V_j} = \sum_{m\in V_j} B^{eT} u_m e^T B^{e} \).

D. Implicit Relation Modeling

Building on our work on learning implicit couplings [22] and its application in inferring implicit rules with item dependencies [33], here, we model and measure the implicit relations between paths in the DPP-based NSP graph \( \mathcal{G} \) in terms of their indirectly linked paths. This measures the implicit relations between NSP patterns in terms of their nonoccurring link sequences. To do this, we introduce the notion of implicit pattern quality terms. On the other hand, some long-size patterns may not share any dependent itemsets, i.e., \( G_{p(y)} = \emptyset \). We further define the maximum IRS.

**Definition 10 (IRS):** Given each shared dependent subsequence \( H \subseteq G_I \), the conditional IRS (CIRS) of \( I \) over \( H \) is defined as the minimum of the iNEMI between each item \( i \) in \( I \) and subsequence \( H \), i.e.,

\[
\text{CIRS}(I|H) = \min_{i \in I} \text{iNEMI}(i, H)
\]

and the IRS of \( I \) is defined as the sum of its CIRS over all dependent itemsets, i.e.,

\[
\text{IRS}(I) = \sum_{H \subseteq G_I} \text{CIRS}(I|H) / |G_I|.
\]

Accordingly, we define the implicit pattern quality of an NSP pattern \( Y_i \).

**Definition 11 (Implicit Pattern Quality):** In \( \mathcal{G} \), the implicit pattern quality of a path \( Y_i \) corresponding to pattern \( Y_i \) is defined as the IRS of the corresponding itemsets transformed from \( Y_i \), denoted as set \( S^i_q(Y_i) \).

However, on the one hand, an NSP pattern may correspond to multiple implicit pattern quality terms. On the other hand, some long-size patterns may not share any dependent itemsets, i.e., \( G_{p(y)} = \emptyset \). We further define the maximum IRS.

**Definition 12 (Maximum IRS):** The maximum IRS of a path \( Y_i \) corresponding to pattern \( Y_i \) is defined as the maximum IRS of the subsets \( S' \) with the largest size, i.e., \( q^m(Y_i) \), calculated as follows:

\[
q^m(Y_i) = \max \left\{ \text{IRS}(S') \left| \left\{ \text{arg max}_{S \subseteq S'} |S| \right. \right. \left. \left. S \subseteq S^i_q(Y_i) \right\} \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \r

\[
\text{IRS}(I) = \sum_{H \subseteq G_I} \text{CIRS}(I|H) / |G_I|.
\]

Furthermore, given an NSP subsequence \( I \), the intersection set of all the dependent subsequence groups of the items in \( I \) constitutes the link groups of \( I \), denoted as \( G_I \equiv \cap_{i \in I} A_i \). Finally, we define the IRS.
given by $B^i_i = q_i(Y_k)\phi_i(Y_k)$, and the dual representation of $L^i$ is constructed as $C^i = B^i B^iT$. Assume that $(\lambda^i_n, \vec{v}^i_n)$ is an eigenvalue/eigenvector pair of $C^i$; then, $(\lambda^i_n, B^iT \vec{v}^i_n)$ is the corresponding pair of $L^i$. Then, (12) is rewritten as follows:

$$P^i(Y) = \frac{1}{e^i_{K,N}} \sum_{|J|=k} \prod_{n \in J} \bar{P}^i(Y) (\lambda^i_{n})_{\bar{J}}.$$

Here, $P^i(Y)$ stands for an elementary DPP with marginal kernel $K^i = \sum_{e \in V^i} B^iT \vec{v}^i_n \vec{v}^i_n B^i$.

E. Overall Relation-Based NSP Selection

To identify the representative NSP subsets, we measure the overall probability of a subset $Y$ by integrating the above explicit and implicit relation-oriented subset probabilities as follows:

$$P^K(Y) = w_e \times P^K_e(Y) + w_i \times P^K_i(Y).$$

$w_e$ and $w_i$ are the parameters to balance between $P^K_e(Y)$ and $P^K_i(Y)$, and $w_e + w_i = 1$. Here, we adopt $w_e = \text{freq}(Y)/\sum_{n \in J} \bar{P}^i(Y)$ and $w_i = \text{freq}(Y)/\sum_{n \in J} \bar{P}^i(Y)$, freq($Y$) is the average frequency of the patterns in $Y$, and $\text{freq}(Y)$ is the average frequency of the flat itemsets transformed from the patterns in $Y$. In this way, the overall probability $P^K(Y)$ is more affected by the major relations of patterns in subset $Y$. Substituting (7) and (13) into (14), we obtain the overall probability of a subset $Y$

$$P^K(Y) = \sum_{d \in \{e,i\}|\omega_{d} \neq 0, J \subseteq \{1,2,\ldots,N_d\}} w_{d} \bar{P}^{d}(Y) \prod_{n \in J} \bar{J}_d^{2}.$$

$P^K(Y)$ will be used to select the $k$-size subset in Section III-F.

F. EINSP Algorithm

EINSP implements the DPP-based actionable NSP discovery process in Fig. 1. EINSP selects a representative NSP subset based on its overall subset probability $P^K(Y)$. Per (15), $P^K(Y)$ is modeled as a mixture of elementwise DPPs (similar to the subset selection proved in [28]). EINSP samples the subset within two major loops corresponding to two phases of sampling in Algorithm 1. In the first loop, a subset of $k$ eigenvectors is selected, where the probability of selecting each eigenvector depends on its associated eigenvalue. A $k$-size index subset $J$ is sampled by $P(J) = \sum_{d \in \{e,i\}|\omega_{d} \neq 0} (\lambda^d_{n})_{\bar{J}} \prod_{n \in J} \bar{J}_d^{2}$, the marginal probability of index $n \in J$ is given as follows:

$$P(n \in J) = \sum_{d \in \{e,i\}|\omega_{d} \neq 0} \frac{\omega_{d} \lambda^d_{n-1}}{\omega_{d} \lambda^d_{n-1}}.$$

With the dual representation, sets $V^e$ and $V^i$ of the eigenvectors of kernels $L^e$ and $L^i$ are represented by their corresponding sets of eigenvectors of $C^e$ and $C^i$. The eigenvectors are denoted as $\hat{V}^e$ and $\hat{V}^i$, with the mapping $V^e = \{B^e B^iT \hat{v}^e \in \hat{V}^i\}$ and $V^i = \{B^iT \hat{v}^i \in \hat{V}^i\}$. Consequently, for any two eigenvectors $\hat{v}^e_n, \hat{v}^i_n \in \hat{V}^e$, we have

Algorithm 1 Pseudocode of EINSP for Actionable NSP Selection

| Input: | Output: |
|---|---|
| $\mathcal{V}$: $\{Y_1, Y_2, \ldots, Y_N\}$, cardinality $k$ | Representative Subset $Y$ |

1. Map $\mathcal{V}$ to a directed graph $G$ per the NSP graph construction.
2. Construct dual representations $C^e$ and $C^i$ and compute their eigenvalue/eigenvector pairs $\{(\lambda^e_n, \hat{v}^e_n)\}_{n=1}^N$ and $\{(\lambda^i_n, \hat{v}^i_n)\}_{n=1}^N$ of $C^e$ and $C^i$, respectively.
3. $J \leftarrow \emptyset$
4. for $n = 1, 2, \ldots, N$ do
5. if $n \sim U[0, 1] < \sum_{d \in \{e,i\}} \omega_{d} \lambda^d_{n}/\omega_{d} \lambda^d_{n-1}$ then
6. $Y \leftarrow Y \cup \{n\}$
7. $k \leftarrow k - 1$
8. if $k = 0$ then break
9. end if
10. end if
11. end for
12. $\hat{V}^e \leftarrow \{v^e_n / \hat{v}^e_n B^e \hat{v}^e_n\}_{n \in J}$
13. $\hat{V}^i \leftarrow \{v^i_n / \hat{v}^i_n B^i \hat{v}^i_n\}_{n \in J}$
14. $Y \leftarrow \emptyset$
15. while $V \neq \emptyset$ do
16. Select $Y_j$ from $\mathcal{Y}$ with $P(Y_j) = \sum_{d \in \{e,i\}} \omega_{d} \bar{P}^d(Y_j) q^d_{|Y_j|} (\hat{v}^e_n B^e \hat{v}^e_n \hat{v}^i_n B^i \hat{v}^i_n)$
17. $Y \leftarrow Y \cup Y_j$
18. $V_e \leftarrow V_{e, \perp}$, where $B^e B_e \hat{v}^e_n \in V_{e, \perp}$ is an orthonormal basis for the subspace of $V_e$ orthogonal to $e_j$
19. $V_i \leftarrow V_{i, \perp}$, where $B^i B_i \hat{v}^i_n \in V_{i, \perp}$ is an orthonormal basis for the subspace of $V_i$ orthogonal to $e_j$
20. end while
21. return $Y$

$\hat{v}^e_n B^e \hat{v}^e_n \hat{v}^i_n B^i \hat{v}^i_n$. Accordingly, the normalization of the vector in $V^e$ and $V^i$ can be computed using only their preimages in $\hat{V}^e$ and $\hat{V}^i$ [28] by updating $\hat{v}^e_n \leftarrow (\hat{v}^e_n B^e \hat{v}^e_n \hat{v}^i_n B^i \hat{v}^i_n)$ and $\hat{v}^i_n \leftarrow (\hat{v}^i_n B^i \hat{v}^i_n \hat{v}^e_n B^e \hat{v}^e_n)$.

In the second phase, a subset $Y$ is generated from the selected eigenvectors. On each iteration of this second loop, the cardinality of $Y$ increases by one, and the dimensionality of $\hat{V}^e$ and $\hat{V}^i$ is reduced by one. Here, $e_j$ is the $j$th standard basis vector, which is all zeros except for a one in the $j$th position. During each iteration, EINSP selects a pattern $Y_j$ according to the following distribution:

$$P(Y_j) = \omega_{d} \bar{P}^d(Y_j) q^d_{|Y_j|} (\hat{v}^e_n B^e \hat{v}^e_n \hat{v}^i_n B^i \hat{v}^i_n).$$

Algorithm 1 summarizes the working mechanism and process of the proposed EINSP method for actionable NSP selection.

IV. COMPLEXITY ANALYSIS AND COMPARISON

Here, we analyze the computational complexity of EINSP and the baseline methods in Section V for comparison. The EINSP computational complexity is determined by its
four constituent parts. First, the NSP graph construction in Section III-B maps an NSP pattern set \( \mathcal{Y} \) to a DPP graph \( \mathcal{G} \); its computational complexity is mainly sensitive to the size of all elements \( E \) in the NSP set \( \mathcal{Y} \) and the pattern number in the set \( |\mathcal{Y}| \), i.e., \( O(|E||\mathcal{Y}|) \). Second, the explicit relation modeling in Section III-C generates a DPP-based graph representation \( C^e \) with the time complexity \( O(|E|^2|\mathcal{Y}|) \); it then generates a subset \( Y \) forming the \( k \)-size SDPP (i.e., k-SDPP) and calculates their probability in (7) by a recursive algorithm [50] with time \( O(|E|k) \). Third, the implicit relation modeling in Section III-D measures the implicit pattern relations sensitive to the item number \( |I| \) and itemset (element) number \( |E| \) with complexity \( O(|I||E|) \); it then calculates the implicit probability of subset \( Y \) in (13) sensitive to items \( I \) and \( k \)-size subsets with complexity \( O(|I||E|) \) (similar to the calculation of explicit probability). Finally, the overall relation-based NSP selection combines the explicit and implicit probability in the second and third steps for NSP subset selection. Consequently, the overall computational complexity of EINSP can be estimated as \( O(|E||\mathcal{Y}| + |E|^2|\mathcal{Y}| + |k||E| + |I||E| + |k||I|) \).

Furthermore, we briefly explain the computational complexity of the baseline methods. First, top-\( K \) selection selects top-\( k \) frequentist patterns from the pattern set \( \mathcal{Y} \) with complexity \( O(|\mathcal{Y}|) \). Second, SAPNSP applies e-NSP with complexity \( O(e-NSP) \) to generate the NSP set with pattern number \( e-NSP \) and then select its subset with complexity \( O(e-NSP) \) (interested readers can refer to [3] for \( O(e-NSP) \)). Third, the k-means method applies k-means clustering to select diverse patterns with complexity \( O(|\mathcal{Y}|^2) \) over the NSP number \( |\mathcal{Y}| \). Finally, the EINSP variant k-SDPP only involves explicit relations with complexity \( O(|E||\mathcal{Y}| + |E|^2|\mathcal{Y}| + |k||E|) \).

The above estimation shows these strategies serve different purposes and are highly incomparable. Their divided computational costs do not reflect their capacity. Empirically comparing their computational costs in experiments does not give much insight into how and why they work, which could instead generate misleading indications. Therefore, in the following evaluation, we focus on evaluating the capacity of the EINSP design in selecting more actionable patterns. EINSP and its baseline strategies are evaluated in terms of their capacity of: 1) pattern (element and item) coverage and diversity (see Section V-A); 2) pattern size/length and diversity clustering effect (see Section V-B); 3) learning implicit pattern relations (see Section V-C); 4) handling data with complex data factor combinations (see Section V-D); 5) ablation study (in Sections V-A–V-D on k-SDPP with EINSP); 6) learning 17 synthetic data with different complexities (see Tables I and II); and 7) learning six highly divided real-life data (see Figs. 2–6). These comprehensive evaluations complement the above complexity analysis with a deep insight into the soundness, robustness, scalability, and flexibility of our approach.

V. EXPERIMENTS AND EVALUATION

The empirical analysis of the proposed EINSP in comparison with four baselines is undertaken on six real-life datasets DS1–DS6 and 17 synthetic datasets. EINSP is evaluated against baselines: Top-k Selection (Top-k), SAPNSP, k-Means Baseline, and k-SDPP. We introduce the datasets and baselines in the Supplemental Material. We then evaluate the EINSP performance in terms of multiple evaluation perspectives.

A. Pattern Coverage

The sequence coverage and item coverage [1], [3] of selected subsets measure the diversity of selected NSPs and distinct NSP items. We report the results of EINSP and the baselines in Sections V-A1 and V-A2. We also compare the average item frequency of selected NSP subsets in Section V-A3. Here, we first quantify the measures sequence coverage, item coverage, and average item frequency for this work.

Sequence coverage measures the diversity of the selected NSP itemsets in the entire sequence database.

Definition 13 (Sequence Coverage): The sequence coverage of subset \( S \) in dataset \( D \) is denoted as \( SC(S|D) \), which is the ratio of the data sequences in \( D \) that covers at least one pattern in \( S \) with respect to the size of the dataset

\[
SC(S|D) = \frac{|\{s_d|s_d \in D \land (\exists s_p \in S \ s.t. s_p \subseteq s_d)\}|}{|D|}
\]  

where \( s_d \) is a data sequence from dataset \( D \), and \( s_p \) is a pattern from the selected subset \( S \).

The sequence coverage \( SC(S|D) \) is always much lower than the sum of the frequency of the patterns in \( S \), i.e., \( SC(S|D) \ll \sum_{s_p \in S} sup(s_p|D) \), \( sup(s_p|D) \) is the support of pattern \( s_p \) in dataset \( D \). This is because the cover sets of different patterns may not be disjoint, i.e., \( cov(s_p|D) \cap cov(s_p'|D) \neq \emptyset \) where \( s_p, s_p' \in S \) and \( cov(s_p|D) = \{s_d|s_d \in D \land s_p \subseteq s_d\} \).

Item coverage measures the diversity of items in the selected NSP set in the entire item set.

Definition 14 (Item Coverage): The item coverage of subset \( S \) in dataset \( D \) measures the ratio of the items in \( D \) covered by at least one pattern in \( S \) with respect to the whole item population, i.e.,

\[
IC(S|D) = \frac{|\{i|\exists s_d \in D, s_p \in S \ s.t. i \subseteq s_d \land i \subseteq s_p\}|}{|I_D|}
\]  

where \( i \) is an item and \( I_D = \{i|\exists s_d \in D \ s.t. i \subseteq s_d\} \) is the set of items covered by dataset \( D \).

The average item frequency measures the frequentist significance of items covered by the selected NSP set.

Definition 15 (Average Item Frequency): The item frequency of an item \( i \) in subset \( S \) is the ratio of its occurrence times in \( S \) with respect to the size of \( S \). Accordingly, the average item frequency in subset \( S \) is defined as

\[
AF(S|D) = \frac{1}{|I_S|} \sum_{i \in I_S} \left|\{s_p|s_p \in S \land i \subseteq s_p\}\right|
\]  

In general, an ideal representative subset has a high sequence coverage such that a large proportion of data sequences in the original dataset are covered by a small-scale subset. A subset with a higher item coverage indicates more informative and diverse items covered in the subset selected from the entire itemset. A subset with a lower average item frequency tends to be more diverse and balanced. These
measures quantify the various levels of diversity of an NSP miner.

1) Sequence Coverage: Fig. 2 shows the results of the sequence coverage of EINSP and the baselines on six real-life datasets. Of all the methods, top-k selection performs worst as it assumes that the covered sets of the selected NSP patterns are disjoint and completely neglects the diversity of the selected subset. This also shows that this method and settings are inapplicable for real-world cases. As a result, top-k selection achieves a relatively lower sequence coverage on datasets with a smaller average length, such as DS1 and DS4, since a larger proportion of data sequences covers multiple short-sized and high-frequency patterns, while the NSPs with relatively rare entities are ignored.

Compared with top-k selection, SAPNSP achieves a slightly better sequence coverage. SAPNSP benefits from considering both the frequency of a pattern and the internal couplings between elements of a pattern in terms of contribution. It thus keeps some long-sized patterns and achieves a higher contribution rate, resulting in a relatively higher diversity of the selected subset. However, the superiority of SAPNSP over top-k selection is very limited because it assumes that the contributions of selected patterns are independent of each other and ignores the contribution ignores the diversity of the selected subset.

Furthermore, top-k selection and SAPNSP lag behind the k-means method because the k-means method selects patterns from each cluster of the NSP collection and guarantees that the covered set of selected patterns shares a much smaller overlap between clusters. Accordingly, the k-means method achieves a much higher sequence coverage on sparse data. By increasing the k value, its superiority becomes more obvious. For example, DS3 is a sparse dataset with 46084 distinct items, the sequence coverage of the k-means method on DS3 improves by 30% over top-k selection and more than 20% over SAPNSP with k = 30, and around 55% over top-k selection and about 46% over SAPNSP with k = 150. These prove the importance of measuring NSP diversity in discovering representative NSP subsets. However, the k-means method is incapable of jointly modeling the quality and diversity of each pattern and does not capture the underlying implicit relations between NSP patterns.

The experiment results indicate that EINSP and its simplified k-SDPP significantly outperform all the other baselines in terms of sequence coverage. Particularly, EINSP achieves much higher sequence coverage compared with the baselines by a maximum of 315%, 36.2%, 224%, 204%, 144%, and 44.9% and an average of 225%, 24.7%, 148%, 149%, 106%, and 34.1% on the datasets for k = 30 for 30-size NSP subsets. The performance improvement made by EINSP on datasets DS2 and DS6 is less significant. When k increases to 150, the improvement on these datasets increases to a maximum of 64.4% and 63.1% and an average of 46.1% and 47.9%, respectively. Overall, both EINSP and k-SDPP show strong superiority over all the baselines on six datasets, showing the effective design of modeling explicit element and pattern couplings in DPP-based NSP representations.
In addition, compared with k-SDPP that only models the explicit relations between elements and between patterns, EINSP jointly models the compound explicit and implicit element/pattern relations in NSPs. EINSP, thus, additionally contributes to an average of 25.4%, 8.00%, 27.4%, 25.9%, 28.3%, and 11.4% performance improvements on the respective datasets over k-SDPP. In comparison with the *k-means* method that only evaluates the importance of each pattern in each cluster in terms of pattern frequency, k-SDPP jointly considers each pattern's quality and diversity in its DPP-based design. k-SDPP, thus, additionally contributes to an average of 186%, 27.9%, 71.7%, 111%, 96.0%, and 32.5% performance improvements on the six datasets over *k-means* baseline. These results show that EINSP captures the NSP pattern diversity, co-occurrence-based element/pattern couplings, and non-co-occurrence-based element/pattern relations, which greatly contributes to much higher sequence coverage of the discovered NSPs. Finally, EINSP and its simplified version k-SDPP achieve better performance and higher coverage with a higher *k*, which shows that EINSP and k-SDPP are scalable with an increase of the parameter *k* in the DPP-based NSP graph.

In summary, our method EINSP and its simplified version k-SDPP achieve much better performance in terms of pattern coverage and diversity. The experiments demonstrate the effectiveness of EINSP's design for capturing diversified and highly probable NSPs. This is achieved by jointly modeling co-occurrence-based explicit element and pattern couplings, and implicit element and pattern relations in selecting representative NSPs.

2) Item Coverage: Fig. 3 shows the item coverage of the subsets selected by EINSP and baselines on the real-life datasets. The results show that both *top-* *k*-selection and SAPNSP achieve the worst item coverage on all datasets and with different *k* values. This is because they always tend to select short-size patterns consisting of only high-frequency items but ignore those rarely observed items.

In contrast, EINSP always achieves the highest item coverage of all the baselines on six datasets, and its selected subset covers patterns with more distinct items. In addition, EINSP always outperforms its simplified k-SDPP by contributing to an average of 18.6%, 22.1%, 19.0%, 18.6%, 28.4%, and 4.50% item coverages on the datasets. This EINSP improvement is made by introducing the implicit element/pattern relations that drive EINSP toward those non-co-occurring patterns with low-frequency item combinations.

Moreover, with the joint consideration of both the quality and diversity of each pattern, k-SDPP beats the *k-means* method by an average of 77.6%, 17.9%, 25.3%, 31.0%, 41.8%, and 33.4% on the six datasets. This demonstrates that capturing both explicit and implicit element/pattern relations can greatly enlarge the item coverage of the selected NSP subset. This especially works for low-frequency items that are usually discarded by frequency and downward closure property-based NSP methods, such as those baselines.

3) Average Item Frequency: A comparison of the average item frequency of the items covered by EINSP and the baselines on all datasets is shown in Fig. 4. The average item frequency of the items covered by EINSP and the property-based NSP methods, such as those baselines, are usually discarded by frequency and downward closure subset. This especially works for low-frequency items that can greatly enlarge the item coverage of the selected NSP capturing both explicit and implicit element/pattern relations.

Fig. 3 shows the top-*k* selection of the items covered by the NSP subsets selected by EINSP and SAPNSP is always much higher than the other three methods. This shows that both *top-* *k*-selection and SAPNSP tend to select patterns combining a smaller number of high-frequency items due to their strong reliance on the frequency and downward closure property-based NSP selection criteria. This results in the high item frequency of selected NSP subsets.

In comparison, all methods considering the diversity of the selected NSP subset achieve a lower average item frequency and, thus, involve more rarely observed items. This may make the findings more diversified, novel, and actionable. In particular, the average item frequency of EINSP-selected NSPs is only 69.8%, 71.1%, 70.1%, 45.3%, 59.8%, and 72.8% of those by *top-* *k*-selection, respectively, on datasets DS1–DS6, and 81.1%, 82.6%, 79.8%, 64.8%, 78.5%, and 85.6% of those by the *k-means* method. These results show that the NSPs selected by EINSP are more balanced and diverse in terms of distinct items covered in the final NSPs.

Combining the experimental findings in Sections V-A2 and V-A2, the NSP subsets selected by EINSP not only cover a larger proportion of distinct items but also contain a higher proportion of those NSPs with relatively low-frequency items. EINSP, thus, produces more balanced, diversified, and novel NSP items and patterns from the entire dataset. In contrast, the baseline methods suffer in these aspects from their limitation in relation to the frequentist-based design. These experiments further show: 1) the importance of modeling not only direct and observable co-occurrence-based element/pattern couplings but also the indirect and implicit couplings between elements and patterns in pattern discovery [22] and 2) the necessity of measuring pattern quality in terms of not only statistical significance but also pattern diversity and novelty in representative NSP discovery.

**B. Average Pattern Size**

The average pattern size of selected subsets measures the selected NSP subset quality since a longer sized pattern.
captures higher long-range dependencies between the elements in the pattern. However, in traditional frequentist-based NSP mining, long-sized patterns are more likely discarded because of their low frequency. Fig. 5 illustrates the average pattern size of EINSP and the baselines on six real-life datasets. First, the average pattern size of top-k selection is always much smaller than that of other methods. Only a small number of its selected patterns contain more than three elements because short-size patterns usually have a higher frequency. This result is consistent with the results of item coverage in Section V-A2.

Second, SAPNSP achieves a slightly bigger average pattern size than top-k selection because SAPNSP applies contribution to pattern selection, which partly favors high-frequency but short-size patterns. However, the downward closure property of contribution indicates that a pattern always has a higher contribution than any of its super sequences, leading to short-sized patterns being more likely selected. Accordingly, the average size of the patterns selected by SAPNSP is always much shorter than that of k-SDPP and EINSP. This shows that SAPNSP cannot discover representative NSPs with long-sized patterns and diversified subsets.

Furthermore, the k-means method achieves a bigger average pattern size than the aforementioned two methods. This is because the k-means method groups patterns into k clusters w.r.t. their diversity and, thus, allows those clusters with long-sized patterns. However, the k-means method only selects the most frequent patterns in a cluster, which are the shortest in size in each belonging cluster.

Contrary to the k-means method, k-SDPP applies explicit pattern quality to evaluate the importance of an NSP pattern, which assigns a high quality to a relatively longer pattern, as shown in (3). Consequently, more long-sized patterns are selected, contributing to the largest average pattern size. As a result, the average pattern size of k-SDPP selections improves by an average of 49.2%, 62.8%, 36.2%, 67.6%, 47.0%, and 53.9% over SAPNSP and 25.5%, 49.6%, 21.1%, 42.2%, 32.1%, and 27.8% over k-means on the respective datasets.

In addition, compared with k-SDPP, EINSP achieves a smaller average pattern size because it uses implicit pattern quality to select NSPs. It assigns a lower IRS to a longer size pattern, similar to [33]. Compared with the baselines, EINSP contributes to an average improvement of 40.2%, 53.5%, 27.9%, 44.9%, 30.3%, and 38.0% over SAPNSP and 17.8%, 40.9%, 13.7%, 22.3%, 17.2%, and 14.3% over the k-means method w.r.t. the average pattern size on the respective datasets. Finally, as seen in Section V-C, introducing implicit pattern relations into EINSP further improves the average IRS of the subset selected by EINSP at the expense of smaller average pattern size. This shows that EINSP selects NSPs that are significant, diverse, and indirectly coupled, which cannot be identified by the existing NSP methods.

C. Average Implicit Pattern Relation Strength

As discussed in Section I, some lowly frequent but highly implicitly coupled patterns may reveal complex but insightful knowledge to inform business decision-making. Accordingly, the average implicit pattern relation strength measures the quality of the subset selected by an NSP miner, with the experiment results shown in Fig. 6.

The three baselines, top-k selection, SAPNSP, and the k-means, always achieve similar but low average implicit pattern relation strength. This is because they do not consider the implicit relations between patterns, i.e., their co-occurrence-based pattern selection ignores indirectly coupled elements and itemsets, and their patterns. In contrast, k-SDPP achieves much higher average implicit pattern relation strength by a maximum of 148.8%, 418.7%, 419.2%, and 375.5% for $k = 30$ and an average of 141.2%, 320.0%, 374.4%, and 354.2% for $k = 150$ on datasets DS3, DS5, DS6, and DS7, respectively. Accordingly, k-SDPP shows an improvement over the average implicit pattern relation strength by a maximum of 66.4%, 214.6%, 148.0%, and 101.9% and an average of 60.1%, 158.0%, 120.7%, and 92.0% on these datasets. These results show that k-SDPP partially involves the indirect couplings between elements and selects those patterns consisting of non-co-occurrence-based elements.

Benefiting from jointly modeling the explicit and implicit relations between elements and between patterns, EINSP achieves a much higher average implicit pattern relation strength than k-SDPP by an additional 52.8%, 21.1%, 63.0%, 75.3%, 74.4%, and 27.3% performance improvements on the six datasets with $k = 30$ and 31.4%, 10.1%, 38.1%, 47.4%, 41.2%, and 13.1% with $k = 150$. This demonstrates the importance of modeling implicit quality and the diversity of non-co-occurring NSP elements and patterns. EINSP, thus, discovers a highly implicitly related subset. Generally speaking, both k-SDPP and EINSP achieve better performance for a larger subset size $k$ in terms of the average implicit pattern relation strength on all datasets but show greater superiority over the baselines for a smaller subset size.

In summary, EINSP achieves significantly bigger pattern coverage, bigger average pattern size, and higher average implicit pattern relation strength of its selected NSP subsets than all of the baselines. The aforementioned experiments conclude the necessity and effectiveness of jointly modeling explicit and implicit element/pattern relations in discovering statistically significant, diverse, and indirectly coupled NSP
pattern sets. Furthermore, in Section V-D, we show the scalability of EINSP in terms of different data factors.

**D. Data Factor Sensitivity**

As shown in [1] and [3], pattern mining methods, including NSP miners, are often sensitive to data characteristics that are quantifiable by data factors, conforming to the results in Sections V-A–V-C for actionable NSP discovery. A reliable NSP miner is expected to maintain its performance superiority over different data factors [3]. Accordingly, here, we empirically test the sensitivity of data factors discussed in Section V-A1 in the Supplemental Material on the sequence coverage of EINSP and the baselines. Table II shows the results of sequence coverage of EINSP on size \(k = 30\), which is always significantly higher than that of the baselines under different data factors. This is consistent with the results of the other comparisons. In addition, the sequence coverage of EINSP is highly sensitive to factors \(C\), \(T\), and \(N\). EINSP works better on the data with higher \(C\), higher \(T\), and lower \(N\). By increasing factor \(C\), more long-sized sequences are generated, and EINSP more likely covers the long-sized patterns, as shown in Section V-B. Moreover, a higher \(C\) leads to a larger number of frequent elements being generated, and the corresponding diversity vector in EINSP contributes to a more diverse subset and makes EINSP more effective.

Table II shows the results of sequence coverage of EINSP on size \(k = 30\), which is always significantly higher than that of the baselines under different data factors. This is consistent with the results of the other comparisons. In addition, the sequence coverage of EINSP is highly sensitive to factors \(C\), \(T\), and \(N\). EINSP works better on the data with higher \(C\), higher \(T\), and lower \(N\). By increasing factor \(C\), more long-sized sequences are generated, and EINSP more likely covers the long-sized patterns, as shown in Section V-B. Moreover, a higher \(C\) leads to a larger number of frequent elements being generated, and the corresponding diversity vector in EINSP contributes to a more diverse subset and makes EINSP more effective.

For \(k = 150\), when \(C\) increases from 6 to 14, the sequence coverage of EINSP grows by 77.3% and outperforms the baselines by an average of 55.8% and a maximum of 73.6%.

### Table I

**Sequence Coverage Sensitivity of Methods Against Data Factors on \(k = 30\)**

| Factors | Dataset Name | Top-k | SAPNSP | k-means | k-SDPP | EINSP |
|---------|--------------|-------|--------|---------|--------|-------|
| C       | C0_T0_S8_I8_DB10k_N0.1k | 14.96% | 15.92% | 16.10% | 23.53% | 32.58% |
|         | C0_T0_S8_DB10k_N0.1k     | 15.99% | 16.65% | 17.88% | 24.84% | 33.67% |
|         | C10_T0_S8_DB10k_N0.1k    | 18.52% | 19.25% | 20.19% | 27.95% | 38.12% |
|         | C12_T0_S8_DB10k_N0.1k    | 20.79% | 21.56% | 23.11% | 31.78% | 42.96% |
|         | C14_T0_S8_DB10k_N0.1k    | 27.37% | 28.39% | 29.59% | 39.65% | 54.78% |
| T       | C0_T0_S8_DB10k_N0.1k     | 14.39% | 15.01% | 15.74% | 21.72% | 30.05% |
|         | C0_T8_S8_DB10k_N0.1k     | 18.52% | 19.25% | 20.19% | 27.95% | 38.12% |
|         | C20_S8_DB10k_N0.1k      | 25.65% | 26.69% | 27.56% | 37.86% | 50.64% |
|         | C10_T0_S8_DB10k_N0.1k    | 28.49% | 29.50% | 30.83% | 42.81% | 56.71% |
|         | C10_T12_S8_DB10k_N0.1k   | 37.19% | 38.14% | 39.76% | 53.93% | 68.25% |

### Table II

**Sequence Coverage Sensitivity of EINSP and Baselines w.r.t. Data Factors for \(k = 150\)**

| Factors | Dataset Name | Top-k | SAPNSP | k-means | k-SDPP | EINSP |
|---------|--------------|-------|--------|---------|--------|-------|
| C       | C0_T0_S8_I8_DB10k_N0.1k | 22.80% | 24.25% | 25.74% | 35.41% | 43.77% |
|         | C0_T0_S8_DB10k_N0.1k     | 26.95% | 28.10% | 29.81% | 40.94% | 49.25% |
|         | C10_T0_S8_DB10k_N0.1k    | 28.67% | 29.74% | 31.05% | 42.04% | 52.09% |
|         | C12_T0_S8_DB10k_N0.1k    | 33.66% | 34.74% | 36.95% | 49.97% | 62.33% |
|         | C14_T0_S8_DB10k_N0.1k    | 44.71% | 46.34% | 48.16% | 63.97% | 77.61% |
| T       | C0_T4_S8_DB10k_N0.1k     | 24.53% | 25.39% | 26.67% | 35.96% | 44.04% |
|         | C0_T6_S8_DB10k_N0.1k     | 28.67% | 29.74% | 31.05% | 42.04% | 52.09% |
|         | C10_T0_S8_DB10k_N0.1k    | 37.50% | 38.66% | 39.85% | 53.64% | 64.68% |
|         | C12_T0_S8_DB10k_N0.1k    | 51.93% | 52.54% | 54.49% | 73.01% | 85.40% |
|         | C14_T0_S8_DB10k_N0.1k    | 58.21% | 59.94% | 61.84% | 77.24% | 89.13% |
|         | C0_T8_S8_DB30k_N0.1k     | 30.19% | 30.97% | 32.95% | 41.74% | 51.93% |
|         | C0_T8_S8_DB50k_N0.1k     | 30.21% | 30.94% | 32.98% | 41.93% | 52.66% |
|         | C0_T8_S8_DB100k_N0.1k    | 30.25% | 30.90% | 32.10% | 41.47% | 52.12% |
|         | C0_T8_S8_DB200k_N0.1k    | 30.26% | 30.96% | 32.12% | 41.46% | 51.73% |
| DB      | C0_T0_S8_DB100k_N0.1k    | 28.67% | 29.74% | 31.05% | 42.04% | 52.09% |
|         | C0_T0_S8_DB100k_N0.2k    | 23.57% | 24.94% | 25.83% | 33.71% | 45.74% |
|         | C0_T0_S8_DB100k_N0.3k    | 22.54% | 23.24% | 24.10% | 32.79% | 41.87% |
|         | C0_T0_S8_DB100k_N0.4k    | 21.46% | 22.77% | 24.59% | 32.84% | 39.50% |
|         | C0_T0_S8_DB100k_N0.5k    | 17.46% | 18.39% | 19.28% | 26.60% | 34.06% |
Furthermore, when factor $T$ grows, the dataset becomes denser, and thus, each pattern selected by EINSP covers more data sequences. For example, when $T$ increases from 4 to 12 for $k = 150$, the sequence coverage of EINSP increases by 85.4% and is 17.0% higher than that of k-SDPP, which is the second best method of all the baselines.

In addition, the growth of factor $DB$ has a limited impact on all the methods because it does not change the distribution of a dataset. Finally, EINSP achieves a lower sequence coverage on those datasets with higher $N$. This is due to the fact that: 1) increasing factor $N$ produces more sparse data sequences, leading to a lower proportion of its sequences being covered by a small-scale subset and 2) less frequent elements are available in a sparse dataset, making the diversity vector in EINSP less effective. In summary, the data factor sensitivity analysis further confirms the design of EINSP for more diverse NSP selection.

VI. DISCUSSION

As discussed in Section I, NSA and NSP minings are often more informative for decision-making, particularly for problems and applications with negative feedback, such as in recommender systems; nonoccurring behaviors and events, such as in medical and health treatments; and risk, safety, and security, matters, such as undeclared behaviors. However, the limited research and available algorithms on NSA prevent its widespread applications.

The nonoccurrence nature of negative sequences hides their great potential for applications. The various combinatorial issues further make NSA and NSP minings much more challenging than PSA and PSP minings, which has been widely explored in areas such as genomic analysis and pattern mining. A critical issue in existing sequence analysis is the underlying frequentist-based pattern selection on the downward closure property held between patterns. This inclicts significantly higher restrictions on NSA than PSA due to the more sophisticated combinatorial scenarios of negative items, elements, itemsets, and patterns.

This article introduces a new way of discovering not only significant but also diverse and indirectly coupled NSPs based on the DPP-based NSP representations. Our work carries the study of pattern relation analysis [19] and coupling learning [22] into NSP mining to disclose the indirect couplings between NSP items, elements, and patterns that do not co-occur. These cannot be achieved by the frequentist and downward closure property.

Accordingly, EINSP opens a new direction, i.e., NSA with pattern selection by analyzing both explicit and implicit pattern relations and element relations, which can be applied to PSA and PSP minings [19]. More theoretical studies are required to explore comprehensive explicit and implicit element/pattern relations and structural relations in NSP pattern relation analysis. Examples are involving item constraint, element constraint, size constraint, format constraint, and combinatorial constraint [1], [3] into the graph-based NSP representations and NSA. Other opportunities lie in theoretical research on analyzing relations between elements and patterns in extremely sparse and diverse but large sequential data.

Last but not least, this work argues the importance of discovering actionable NSPs that are significant, diverse, novel, and informative by involving NSP statistics, diversity, and informativeness in terms of element and pattern couplings.

Very limited research is available on pattern actionability and actionable knowledge discovery, which are increasingly essential in discovering actionable intelligence for AI to inform decision-making.

VII. CONCLUSION

While NSA has been rarely studied, it has played a strong role in discovering significant occurring and nonoccurring entities, events, and behaviors. Many significant theoretical and practical challenges surround NSA, e.g., a lack of theoretical foundation, complicated combinatorial scenarios, nonoccurring relations, and extreme computational cost.

This work represents a new direction in NSA to discover representative NSPs that are of high quality, diversity, and informativeness, i.e., toward actionable NSP discovery. We propose a novel DPP-based NSP graph representation and the DPP-based EINSP models for both explicit and implicit relations in positive and negative sequential elements and patterns in terms of the co-occurring and nonoccurring probabilistic distributions over all possible subsets in the pattern cohort, EINSP selects a representative NSP subset composed of high-quality and diverse NSP elements and patterns. Such actionable NSP discovery deserves further research with various potentials and challenges.

Our work is also new in proposing DPP-based NSP graph representation and NSP pattern relation analysis. Both theoretical and empirical analyses demonstrate the strong potential of EINSP in discovering more representative and informative patterns with wider coverage, larger diversity, and higher quality in terms of different data factors. The NSP graph representation and pattern relation analysis open various opportunities for NSA and general sequence analysis.

REFERENCES

[1] W. Wang and L. Cao, “Negative sequence analysis: A review,” ACM Comput. Surv., vol. 52, no. 2, p. 32:1–32:39, 2019.
[2] L. Cao, P. S. Yu, and V. Kumar, “Nonoccuring behavior analytics: A new area,” IEEE Intel. Syst., vol. 30, no. 6, pp. 4–11, Nov. 2015.
[3] L. Cao, X. Dong, and Z. Zheng, “E-NSP: Efficient negative sequential pattern mining,” Artif. Intell., vol. 235, pp. 156–182, Jun. 2016.
[4] N. P. Lin, H.-J. Chen, and W.-H. Hao, “Mining negative sequential patterns,” in Proc. ICACS, 2007, pp. 654–658.
[5] S.-C. Hsueh, M.-Y. Lin, and C.-L. Chen, “Mining negative sequential patterns for E-commerce recommendations,” in Proc. IEEE Asia-Pacific Services Comput. Conf., Dec. 2008, pp. 1213–1218.
[6] L. Cao, Y. Zhao, and C. Zhang, “Mining impact-targeted activity patterns in imbalanced data,” IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1053–1066, Aug. 2008.
[7] E. Spyropoulou and T. De Bie, “Mining approximate multi-relationship patterns,” in Proc. Int. Conf. Data Sci. Adv. Anal. (DSAA), Oct. 2014, pp. 477–483.
[8] R. Agrawal and R. Srikant, “Mining sequential patterns,” in Proc. ICDE, 1995, pp. 3–14.
[9] R. U. Kiran, Y. Watanobe, B. Chaudhury, K. Zettu, M. Toyoda, and M. Kitsuregawa, “Discovering maximal periodic-frequent patterns in very large temporal databases,” in Proc. IEEE 7th Int. Conf. Data Sci. Adv. Anal. (DSAA), Oct. 2020, pp. 1–10.
[10] B. Bez and A. R. Butt, “Fixed point for set-valued mappings satisfying an implicit relation in partially ordered metric spaces,” Nonlinear Anal., Theory, Methods Appl., vol. 71, no. 9, pp. 3699–3704, Nov. 2009.
