(resp. lower) than the probability of any subsequence.

Proposition 4.1: Consider two sequences S_1 and S_2 occurring in an observation sequence. If S_1 is a subsequence of S_2, then $\mathcal{P}_T(S_1) \geq \mathcal{P}_T(S_2)$ and $\mathcal{P}_P(S_1) \leq \mathcal{P}_P(S_2)$.

We now define totally and partially unexplained sequences.

Definition 4.5 (Unexplained sequences): Let v be an observation sequence, $\tau \in [0, 1]$ a probability threshold, and $L \in \mathbb{N}^+$ a length threshold. A sequence S occuring in v is:

- A *totally unexplained sequence* if (i) $\mathcal{P}_T(S) \geq \tau$, (ii) $|S| \geq L$, and (iii) S is maximal, i.e., there is no sequence $S' \neq S$ occurring in v s.t. S is a subsequence of S', $\mathcal{P}_T(S') \geq \tau$, and $|S'| \geq L$.
- A *partially unexplained sequence* if (i) $\mathcal{P}_P(S) \geq \tau$, (ii) $|S| \geq L$, and (iii) S is minimal, i.e., there is no sequence $S' \neq S$ occurring in v s.t. S' is a subsequence of S, $\mathcal{P}_P(S') \geq \tau$, and $|S'| \geq L$.

In this definition, L is the minimum length a sequence must be for it to be considered possibly unexplained. Totally unexplained sequences (TUSs for short) S have to be maximal because once we find S, any sub-sequence of it is (totally) unexplained with probability greater than or equal to that of S. On the other hand, partially unexplained sequences (PUSs for short) S' have to be minimal because once we find S', any super-sequence of it is (partially) unexplained with probability greater than or equal to that of S'.

Intuitively, an unexplained sequence is a sequence of action symbols that are observed in the observation sequence and poorly explained by known activity models. Such sequences might correspond to unknown variants of known activities or to entirely new—and unknown—activities.

An *Unexplained Sequence Problem* (USP) instance is a triple $I = \langle v, \tau, L \rangle$ where v is an observation sequence, $\tau \in [0, 1]$ is a probability threshold, and $L \in \mathbb{N}^+$ is a length threshold. We want to find the sets $\mathcal{A}^{tu}(I)$ and $\mathcal{A}^{pu}(I)$ of all totally and partially unexplained sequences, respectively. When I is clear from context, we will drop it.

The following definition introduces the top-k totally and partially unexplained sequences. Intuitively, these are k unexplained sequences having maximum probability.

Definition 4.6 (Top-k unexplained sequences): Consider a USP instance and let $k \in \mathbb{N}^+$. $\mathcal{A}^{tu}_k \subseteq \mathcal{A}^{tu}$ (resp. $\mathcal{A}^{pu}_k \subseteq \mathcal{A}^{pu}$) is a set of top-$k$ totally (resp. partially) unexplained sequences iff $|\mathcal{A}^{tu}_k| = \min\{k, |\mathcal{A}^{tu}|\}$ (resp. $|\mathcal{A}^{pu}_k| = \min\{k, |\mathcal{A}^{pu}|\}$), and $\forall S \in \mathcal{A}^{tu}_k, \forall S' \in \mathcal{A}^{tu} - \mathcal{A}^{tu}_k$ (resp. $\forall S \in \mathcal{A}^{pu}_k, \forall S' \in \mathcal{A}^{pu} - \mathcal{A}^{pu}_k$) $\mathcal{P}_T(S) \geq \mathcal{P}_T(S')$ (resp. $\mathcal{P}_P(S) \geq \mathcal{P}_P(S')$).

Suppose we have a USP instance. For any $S, S' \in \mathcal{A}^{tu}$ (resp. $S, S' \in \mathcal{A}^{pu}$), we write $S =_T S'$ (resp. $S =_P S'$) iff $\mathcal{P}_T(S) = \mathcal{P}_T(S')$ (resp. $\mathcal{P}_P(S) = \mathcal{P}_P(S')$). Obviously, $=_T$ (resp. $=_P$) is an equivalence relation and determines a set $\mathcal{C}^{tu}$ (resp. $\mathcal{C}^{pu}$) of equivalence classes. For any equivalence class $C \in \mathcal{C}^{tu}$ (resp. $C \in \mathcal{C}^{pu}$) we define $\mathcal{P}_T(C)$ (resp. $\mathcal{P}_P(C)$) as the (unique) probability of the sequences in C.

Symbol	Description
$\mathcal{A}$	Set of stochastic activities
s	Action symbol
f	Observation ID (OID)
$f.ts$	Timestamp associated with observation ID f
$f.obs$	Set of action symbols associated with observation ID f
v	Observation sequence
o and $\mathcal{O}$	Activity occurrence and set of activity occurrences
w and $\mathcal{W}$	Possible world and set of possible worlds
$\langle v_1, \ldots, v_m \rangle$	Conflict-based partitioning (CBP) of observation sequence v. Each v_i is called a *segment*
$NLC(v)$	Set of non-linear constraints for observation sequence v
$LC(v)$	Set of linear constraints for observation sequence v
$w \not\models_T S$	Sequence S is totally unexplained in world w
$w \not\models_P S$	Sequence S is partially unexplained in world w
$\mathcal{I}_T(S)$	Probability interval that sequence S is totally unexplained
$\mathcal{I}_P(S)$	Probability interval that sequence S is partially unexplained
$\mathcal{P}_T(S)$	(Point) Probability that sequence S is totally unexplained
$\mathcal{P}_P(S)$	(Point) Probability that sequence S is partially unexplained

TABLE 1: Notation

Compared with the top-k unexplained sequences, the top$-k$ unexplained classes find *all* the unexplained sequences having the k highest probabilities.

Definition 4.7 (Top-k unexplained classes): Consider a USP instance and let $k \in \mathbb{N}^+$. $\mathcal{C}^{tu}_k \subseteq \mathcal{C}^{tu}$ (resp. $\mathcal{C}^{pu}_k \subseteq \mathcal{C}^{pu}$) is the set of top-$k$ totally (resp. partially) unexplained classes iff $|\mathcal{C}^{tu}_k| = \min\{k, |\mathcal{C}^{tu}|\}$ (resp. $|\mathcal{C}^{pu}_k| = \min\{k, |\mathcal{C}^{pu}|\}$), and $\forall C \in \mathcal{C}^{tu}_k, \forall C' \in \mathcal{C}^{tu} - \mathcal{C}^{tu}_k$ (resp. $\forall C \in \mathcal{C}^{pu}_k, \forall C' \in \mathcal{C}^{pu} - \mathcal{C}^{pu}_k$) $\mathcal{P}_T(C) > \mathcal{P}_T(C')$ (resp. $\mathcal{P}_P(C) > \mathcal{P}_P(C')$).

Table 1 summarizes the main notation used in the paper.

5 PROPERTIES OF USPS

This section derives properties that can be leveraged (in the next section) to devise efficient algorithms to solve USPs. We first show an interesting property concerning the solution of $NLC(v)$ (some later results rely on it); the following two subsections consider specific properties for totally and partially unexplained sequences.

For a given observation sequence v, we show that if $\langle v_1, \ldots, v_m \rangle$ is a CBP, then we can find solutions of the *non-linear* constraints $NLC(v)$ by solving m *smaller sets of linear constraints.*[6] Let $LC(v)$ be the set of linear constraints of $NLC(v)$ (i.e., all constraints of Definition 4.3 except for the last kind). Henceforth, we use $\mathcal{W}$ to denote $\mathcal{W}(v)$ and $\mathcal{W}_i$ to denote $\mathcal{W}(v_i)$, $1 \leq i \leq m$. A solution of $NLC(v)$ is a mapping $\mathcal{P} : \mathcal{W} \rightarrow [0, 1]$ which satisfies $NLC(v)$. Likewise, a solution of $LC(v_i)$ is a mapping $\mathcal{P}_i : \mathcal{W}_i \rightarrow [0, 1]$ which satisfies $LC(v_i)$. It is important to note that $\mathcal{W} = \{w_1 \cup \ldots \cup w_m \mid w_i \in \mathcal{W}_i, 1 \leq i \leq m\}$.

Theorem 1: Let v be an observation sequence and $\langle v_1, \ldots, v_m \rangle$ a CBP. $\mathcal{P}$ is a solution of $NLC(v)$ iff $\forall i \in [1, m]$ there exists a solution $\mathcal{P}_i$ of $LC(v_i)$

[6] This yields two benefits: (i) It allows us to solve a smaller set of constraints. (ii) It allows us to solve linear constraints which are easier to solve than nonlinear ones. Moreover, it allows us to drastically reduce the space of possible worlds considered, as we can consider each segment v_i (and its corresponding possible worlds) individually, thereby avoiding the blow up we would get by combining possible worlds of different segments. This also applies to Theorems 2 and 4.

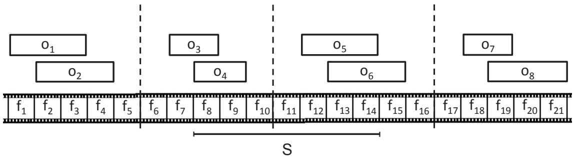

Fig. 5: Conflict-Based Partitioning of a video

s.t. $\mathcal{P}(\bigcup_{i=1}^{m} w_i) = \prod_{i=1}^{m} \mathcal{P}_i(w_i)$ for every $w_1 \in \mathcal{W}_1, \ldots, w_m \in \mathcal{W}_m$.

The following example illustrates the previous theorem.

Example 5.1 (Video example): Consider the video v of Example 4.3 (cf. Figure 3). As shown in Example 4.4, one possible CBP of v is $\langle v_1, v_2 \rangle$, where $v_1 = \langle f_1, \ldots, f_9 \rangle$ and $v_2 = \langle f_{10}, \ldots, f_{16} \rangle$. Theorem 1 says that for each solution $\mathcal{P}$ of $NLC(v)$, there is a solution $\mathcal{P}_1$ of $LC(v_1)$ and a solution $\mathcal{P}_2$ of $LC(v_2)$ s.t. $\mathcal{P}(w_1 \cup w_2) = \mathcal{P}_1(w_1) \times \mathcal{P}(w_2)$ for every $w_1 \in \mathcal{W}_1, w_2 \in \mathcal{W}_2$, and vice versa.

Consider an observation sequence v and let $\langle v_1, \ldots, v_m \rangle$ be a CBP. Given a sequence $S = \langle (f_1, s_1), \ldots, (f_q, s_q) \rangle$ occurring in v, we say that $v_i, v_{i+1}, \ldots, v_{i+n}$ $(1 \le i \le i+n \le m)$ are the segments *containing* S iff $f_1 \in v_i$ and $f_q \in v_{i+n}$. In other words, S spans the segments $v_i, v_{i+1}, \ldots, v_{i+n}$: it starts at a point in segment v_i (as v_i contains the first OID of S) and ends at some point in segment v_{i+n} (as v_{i+n} contains the last OID of S). S_k denotes the *projection* of S on the k-th segment v_k $(i \le k \le i+n)$, that is, the subsequence of S containing all the pairs $(f, s) \in S$ with $f \in v_k$.

Example 5.2 (Video example): Suppose we have a video $v = \langle f_1, \ldots, f_{21} \rangle$ such that $f_i.obs = \{s_i\}$ for $1 \le i \le 21$. In addition, suppose 8 occurrences are detected as shown in Figure 5. Consider the CBP $\langle v_1, v_2, v_3, v_4 \rangle$, where $v_1 = \{f_1, \ldots, f_5\}$, $v_2 = \{f_6, \ldots, f_{10}\}$, $v_3 = \{f_{11}, \ldots, f_{16}\}$, and $v_4 = \{f_{17}, \ldots, f_{21}\}$. Consider now the sequence $S = \langle (f_8, s_8), \ldots, (f_{14}, s_{14}) \rangle$ occurring in v. Then, v_2 and v_3 are the segments containing S. Moreover, S_2 denotes $\langle (f_8, s_8), \ldots, (f_{10}, s_{10}) \rangle$, and S_3 denotes $\langle (f_{11}, s_{11}), \ldots, (f_{14}, s_{14}) \rangle$.

5.1 Totally unexplained sequences

The following theorem says that we can compute $\mathcal{I}_T(S)$ by solving LC (which are linear constraints) for each segment containing S (instead of solving a non-linear set of constraints for the whole observation sequence).

Theorem 2: Consider an observation sequence v. Let $\langle v_1, \ldots, v_m \rangle$ be a CBP and $\langle v_i, \ldots, v_{i+n} \rangle$ the segments containing a sequence S occurring in v. For $i \le k \le i+n$, let

$$\begin{aligned} l_k = \mathbf{minimize} & \textstyle\sum_{w \in \mathcal{W}_k \ s.t. \ w \not\models_T S_k} p_w \\ & \mathbf{subject\ to}\ LC(v_k) \\ u_k = \mathbf{maximize} & \textstyle\sum_{w \in \mathcal{W}_k \ s.t. \ w \not\models_T S_k} p_w \\ & \mathbf{subject\ to}\ LC(v_k) \end{aligned}$$

If $\mathcal{I}_T(S) = [l, u]$, then $l = \prod_{k=i}^{i+n} l_k$ and $u = \prod_{k=i}^{i+n} u_k$.

The following example illustrates the theorem above.

Example 5.3 (Video example): Consider Example 5.2, which is depicted in Figure 5. $\mathcal{I}_T(S)$ can be computed by solving the non-linear program of Definition 4.4 for the whole video v. But Theorem 2 says that $\mathcal{I}_T(S)$ can be computed as $\mathcal{I}_T(S) = [l_2 \times l_3, u_2 \times u_3]$, where l_2, u_2, l_3, u_3 are computed as defined in Theorem 2, i.e. by solving two smaller linear programs for v_2 and v_3.

The following theorem provides a sufficient condition for a pair (f, s) not to be included in any sequence S occurring in v and having $\mathcal{P}_T(S) \ge \tau$.

Theorem 3: Let $\langle v, \tau, L \rangle$ be a USP instance. Given (f, s) s.t. $f \in v$ and $s \in f.obs$, let $\varepsilon = \sum_{o \in \mathcal{O} \ s.t. \ (f,s) \in o} p^*(o) \cdot \frac{Weight(o)}{\sum_{o_j \in C(o)} Weight(o_j)}$. If $\varepsilon > 1 - \tau$, then there does not exist a sequence S occurring in v s.t. $(f, s) \in S$ and $\mathcal{P}_T(S) \ge \tau$.

If the above condition holds for a pair (f, s), then we say that (f, s) is *sufficiently explained. Note that to check whether a pair (f, s) is sufficiently explained, we do not need to solve any set of linear or non-linear constraints, since ε is computed by simply summing the (weighted) probabilities of the occurrences containing (f, s)*. Thus, this result yields a further efficiency. An OID f is *sufficiently explained* iff (f, s) is sufficiently explained for every $s \in f.obs$. If (f, s) is sufficiently explained, then it can be disregarded when identifying unexplained sequences. Moreover, this may allow us to disregard entire parts of observation sequences as shown in the example below.

Example 5.4 (Video example): Consider a USP instance $\langle v, \tau, L \rangle$ where $v = \langle f_1, \ldots, f_9 \rangle$ is s.t. $f_i.obs = \{s_i\}$ for $1 \le i \le 9$, as depicted in Figure 6.

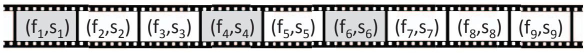

Fig. 6: Sufficiently explained frames in a video.

Suppose $L = 3$ and (f_1, s_1), (f_4, s_4), (f_6, s_6) are sufficiently explained. Even though the theorem is applicable to only a few (f_i, s_i) pairs, we see that no unexplained sequence can be found before f_7 as $L = 3$.

Given a USP instance $I = \langle v, \tau, L \rangle$ and a subsequence v' of v, v' is *relevant* iff (i) v' is a contiguous subsequence of v (ii) $|v'| \ge L$, (iii) $\forall f \in v'$, f is not sufficiently explained, and (iv) v' is maximal (i.e., there does not exist $v'' \ne v'$ s.t. v' is a subsequence of v'' and v'' satisfies (i), (ii), (iii)). We use $relevant(I)$ to denote the set of relevant observation subsequences.

Theorem 3 entails that relevant observation subsequences can be individually considered when looking for totally unexplained sequences because there is no totally unexplained sequence spanning two different relevant observation subsequences.

5.2 Partially unexplained sequences

The following theorem states that we can compute $\mathcal{I}_P(S)$ by solving NLC for the observation subsequence consisting of the segments containing S (instead of solving NLC for the whole observation sequence).

Theorem 4: Consider an observation sequence v. Let $\langle v_1, \ldots, v_m \rangle$ be a CBP and $\langle v_i, \ldots, v_{i+n} \rangle$ be the segments containing a sequence S occurring in v. Let $v^* = v_i \cdot \ldots \cdot v_{i+n}$. $\mathcal{I}_P(S)$ computed w.r.t. v is equal to $\mathcal{I}_P(S)$ computed w.r.t. v^*.

We now illustrate the use of the preceding theorem.

Example 5.5 (Video example): Consider Example 5.2 as shown in Figure 5. By definition, $\mathcal{I}_P(S)$ can be computed by solving the non-linear program of Definition 4.4 for the whole video v. Alternatively, Theorem 4 says that $\mathcal{I}_P(S)$ can be computed by solving the non-linear program of Definition 4.4 for the sub-video $v^* = v_2 \cdot v_3$.

6 TOP-k ALGORITHMS

We now present algorithms to find top-k totally and partially unexplained sequences and classes. For ease of presentation, we assume $|f.obs| = 1$ for every OID f in an observation sequence (this makes the algorithms much more concise – generalization to the case of multiple action symbols per OID is straightforward[7]). Given an observation sequence $v = \langle f_1, \ldots, f_n \rangle$, we use $v(i, j)$ $(1 \leq i \leq j \leq n)$ to denote the sequence $S = \langle (f_i, s_i), \ldots, (f_j, s_j) \rangle$, where s_k is the only element in $f_k.obs$, $i \leq k \leq j$.

6.1 Top-k TUS and TUC

The Top-k TUS algorithm computes a set of top-k totally unexplained sequences in an observation sequence. Note that:

- at every time, $lowest$ is defined as follows:

$$lowest = \begin{cases} -1 & \text{if } |TopSol| < k \\ \min\{\mathcal{P}_T(S) \mid S \in TopSol\} & \text{if } |TopSol| = k \end{cases}$$

- On line 30, "Add S to $TopSol$" works as follows:
  - If $|TopSol| < k$, then S is added to $TopSol$;
  - otherwise, a sequence S' in $TopSol$ having minimum $\mathcal{P}_T(S')$ is replaced by S.

Algorithm 1 Top-k TUS

```
Input: USP instance $I = \langle v, \tau, L \rangle$, $k \geq 1$
Output: Top-$k$ totally unexplained sequences
 1: $TopSol = \emptyset$
 2: for all $v' \in relevant(I)$ do
 3:     $start = 1$;  $end = L$
 4:     repeat
 5:         if $\mathcal{P}_T(v'(start, end)) \geq \tau \wedge \mathcal{P}_T(v'(start, end)) > lowest$ then
 6:             $end' = end$
 7:             while $end < |v'|$ do
 8:                 $end = \min\{end + L, |v'|\}$
 9:                 if $\mathcal{P}_T(v'(start, end)) < \tau$ then
10:                     break
11:                 else
12:                     if $\mathcal{P}_T(v'(start, end)) \leq lowest$ then
13:                         $end = end + 1$
14:                         go to line 33
15:             $s = \max\{end - L, end'\}$;  $e = end$
16:             while $e \neq s$ do
17:                 $mid = \lceil (s + e)/2 \rceil$
18:                 if $\mathcal{P}_T(v'(start, mid)) \geq \tau$ then
19:                     if $\mathcal{P}_T(v'(start, mid)) \leq lowest$ then
20:                         $end = mid + 1$
21:                         go to line 33
22:                     else
23:                         $s = mid$
24:                 else
25:                     $e = mid - 1$
26:             if $start > 1 \wedge \mathcal{P}_T(v'(start - 1, s)) \geq \tau$ then
27:                 $end = s + 1$
28:                 go to line 33
29:             else
30:                 $S = v'(start, s)$;  Add $S$ to $TopSol$
31:                 $start = start + 1$;  $end = s + 1$
32:         else
33:             $start = start + 1$;   $end = \max\{end, start + L - 1\}$
34:     until $end > |v'|$
35: return $TopSol$
```

Leveraging Theorem 3, Top-k TUS considers only relevant observation subsequences of v individually (line 2). When it finds a sequence $v'(start, end)$ of length at least L having a probability of being totally unexplained greater than $lowest$ (line 5), it makes the sequence maximal by adding OIDs on the right (lines 7–14). Instead of adding one OID at a time, $v'(start, end)$ is extended by L OIDs at a time until its probability drops below τ (lines 9–10); a binary search is then performed to find the exact maximum length of the unexplained sequence (lines 15–25). While making the sequence maximal, if the algorithm realizes that the unexplained sequence will not have a probability greater than $lowest$ (i.e., the sequence is not a top-k TUS), then the sequence is disregarded and the process of making the sequence maximal is aborted (lines 12–14 and 19–21). This pruning allows the algorithm to move forward in the observation sequence avoiding computing the exact ending OID of the TUS thereby saving time. Throughout the algorithm, $\mathcal{P}_T$ is computed by applying Theorem 2.

Theorem 5: Algorithm Top-k TUS returns a set of top-k totally unexplained sequences of the input instance.

Algorithm Top-k TUC modifies Top-k TUS as follows to compute the top-k totally unexplained classes:

- At every time, $lowest$ is defined as follows:

$$lowest = \begin{cases} -1 & \text{if } |TopSol| < k \\ \min\{\mathcal{P}_T(C) \mid C \in TopSol\} & \text{if } |TopSol| = k \end{cases}$$

- "Add S to $TopSol$" (line 30) works as follows:
  - If there exists $C \in TopSol$ s.t. $\mathcal{P}_T(C) = \mathcal{P}_T(S)$, then S is added to C;
  - else if $|TopSol| < k$, then the class $\{S\}$ is added to $TopSol$;
  - otherwise the class C in $TopSol$ having minimum $\mathcal{P}_T(C)$ is replaced with $\{S\}$.
- On line 5, $\mathcal{P}_T(v'(start, end)) > lowest$ is replaced with $\mathcal{P}_T(v'(start, end)) \geq lowest$;
- On line 12, $\mathcal{P}_T(v'(start, end)) \leq lowest$ is replaced with $\mathcal{P}_T(v'(start, end)) < lowest$;
- On line 19, $\mathcal{P}_T(v'(start, mid)) \leq lowest$ is replaced with $\mathcal{P}_T(v'(start, mid)) < lowest$;

[7] It suffices to consider the different sequences given by the different action symbols.

Algorithm 2 Top-k PUS

```
Input: USP instance $I = \langle v, \tau, L \rangle$, $k \geq 1$
Output: Top-$k$ partially unexplained sequences
 1: $TopSol = \emptyset$;   $start = 1$;   $end = L$
 2: while $end \leq |v|$ do
 3:   if $\mathcal{P}_P(v(start, end)) < \tau$ then
 4:     $end' = end$
 5:     while $end < |v|$ do
 6:       $end = \min\{end + L, |v|\}$
 7:       if $\mathcal{P}_P(v(start, end)) \geq \tau$ then
 8:         break
 9:     if $\mathcal{P}_P(v(start, end)) \geq \tau$ then
10:       if $\mathcal{P}_P(v(start, end)) > lowest$ then
11:         $s = \max\{end' + 1, end - L + 1\}$;   $e = end$
12:         while $e \neq s$ do
13:           $mid = \lfloor (s + e)/2 \rfloor$
14:           if $\mathcal{P}_P(v(start, mid)) < \tau$ then
15:             $s = mid + 1$
16:           else
17:             if $\mathcal{P}_P(v(start, mid)) \leq lowest$ then
18:               $start = start + 1$;   $end = mid + 1$
19:               go to line 2
20:             else
21:               $e = mid$
22:         $end = e$
23:       else
24:         $start = start + 1$;   $end = end + 1$
25:         go to line 2
26:     else
27:       return $TopSol$
28:   $s' = start$;  $e' = end - L + 1$
29:   while $e' \neq s'$ do
30:     $mid = \lceil (s' + e')/2 \rceil$
31:     if $\mathcal{P}_P(v(mid, end)) < \tau$ then
32:       $e' = mid - 1$
33:     else
34:       if $\mathcal{P}_P(v(mid, end)) \leq lowest$ then
35:         $start = mid + 1$;   $end = end + 1$
36:         go to line 2
37:       else
38:         $s' = mid$
39:   if $\mathcal{P}_P(v(s', end - 1)) \geq \tau \wedge |v(s', end - 1)| \geq L$ then
40:     $start = s' + 1$;   $end = end + 1$
41:     go to line 2
42:   else
43:     $S = v(s', end)$;  Add $S$ to $TopSol$
44:     $start = s' + 1$;   $end = end + 1$
45: return $TopSol$
```

The algorithm obtained by applying the modifications above is named Top-k TUC.

Theorem 6: Algorithm Top-k TUC returns the top-k totally unexplained classes of the input instance.

6.2 Top-k PUS and PUC

The Top-k PUS algorithm below computes a set of top-k partially unexplained sequences in an observation sequence. Note that:

- at each time, $lowest$ is defined as follows:

$$lowest = \begin{cases} -1 & \text{if } |TopSol| < k \\ \min\{\mathcal{P}_P(S) \mid S \in TopSol\} & \text{if } |TopSol| = k \end{cases}$$

- On line 43, "Add S to $TopSol$" works as follows:
  - If $|TopSol| < k$, then S is added to $TopSol$;
  - otherwise, a sequence in $TopSol$ having minimum $\mathcal{P}_P$ is replaced by S.

To find an unexplained sequence, Algorithm Top-k PUS starts with a sequence of length at least L and adds OIDs to its right until its probability of being partially unexplained is above the threshold. As in the case of Top-k TUS, this is done by adding L OIDs at a time (lines 5–8) and then performing a binary search (lines 9–27). When performing the binary search, if at some point the algorithm realizes that the partially unexplained sequence will not have a probability greater than $lowest$, then the sequence is disregarded and the binary search is aborted (lines 17–19 and lines 24–25). Otherwise, the sequence is shortened on the left making it minimal (lines 28–38) by performing a binary search instead of proceeding one OID at a time. If the algorithm realizes that the partially unexplained sequence will not have a probability greater than $lowest$, then the sequence is disregarded and the shortening process is aborted (lines 34–36). This allows the algorithm to avoid computing the exact starting OID of the PUS, thus saving time. Note that $\mathcal{P}_P$ is computed by applying Theorem 4.

Theorem 7: Algorithm Top-k PUS returns the set of top-k partially unexplained sequences of the input instance.

Algorithm Top-k PUC modifies Top-k PUS as follows to compute the top-k partially unexplained classes:

- At every time, $lowest$ is defined as follows:

$$lowest = \begin{cases} -1 & \text{if } |TopSol| < k \\ \min\{\mathcal{P}_P(C) \mid C \in TopSol\} & \text{if } |TopSol| = k \end{cases}$$

- "Add S to $TopSol$" (line 43) works as follows:
  - If there exists $C \in TopSol$ s.t. $\mathcal{P}_P(C) = \mathcal{P}_P(S)$, then S is added to C;
  - else if $|TopSol| < k$, then the class $\{S\}$ is added to $TopSol$;
  - otherwise the class C in $TopSol$ having minimum $\mathcal{P}_P(C)$ is replaced with $\{S\}$.
- On line 10, $\mathcal{P}_P(v(start, end)) > lowest$ is replaced with $\mathcal{P}_P(v(start, end)) \geq lowest$;
- On line 17, $\mathcal{P}_P(v(start, mid)) \leq lowest$ is replaced with $\mathcal{P}_P(v(start, mid)) < lowest$;
- On line 34, $\mathcal{P}_P(v(mid, end)) \leq lowest$ is replaced with $\mathcal{P}_P(v(mid, end)) < lowest$;

The algorithm obtained by applying the modifications above is named Top-k PUC.

Theorem 8: Algorithm Top-k PUC returns the top-k partially unexplained classes of the input instance.

7 Experimental Evaluation

We implemented Algorithms Top-k TUS, Top-k PUS, Top-k TUC and Top-k PUC, and experimentally evaluated both running time and accuracy on real-world datasets video and cyber security datasets.

7.1 Video Surveillance Domain

We evaluated our framework on two video datasets: (i) a video we shot by monitoring a university parking lot, and (ii) a benchmark dataset about video surveillance in an airport [41]. The frame observations have been generated in