New Results on Listing Closed Sets of Strongly Accessible Set Systems (Extended Abstract)

Mario Boley[a] MARIO.BOLEY@IAIS.FRAUNHOFER.DE
Tamás Horváth[b,a] TAMAS.HORVATH@IAIS.FRAUNHOFER.DE
Axel Poigné[a] AXEL.POIGNE@IAIS.FRAUNHOFER.DE
Stefan Wrobel[a,b] STEFAN.WROBEL@IAIS.FRAUNHOFER.DE

[a]Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, Germany
[b]Department of Computer Science, University of Bonn, Germany

1. Introduction

We study the problem of listing all closed sets (i.e., fixpoints) of a closure operator $\sigma : \mathcal{F} \to \mathcal{F}$, where $\mathcal{F}$ is a subset of the power set $\mathcal{P}(E)$ for some finite ground set E. We assume that $\mathcal{F}$ is given by a membership oracle. This problem appears for instance in discrete mathematics, formal concept analysis, and data mining. Except for data mining, most of the results are restricted to the case $\mathcal{F} = \mathcal{P}(E)$, for which several efficient algorithms have been developed. In data mining, there is a special interest in listing all closed sets satisfying some additional "interestingness" constraint (e.g., frequency). Such constraints usually specify an independence system, i.e., a set system closed under subsets. The case that $\mathcal{F}$ is an independence system can easily be reduced to the case $\mathcal{F} = \mathcal{P}(E)$.

In a former work (Boley et al., 2007) we defined different graph mining settings raising the more general problem of listing all closed sets of *strongly accessible* set systems. In a nutshell, strong accessibility means that every $Y \in \mathcal{F}$ can be reached from all $X \subset Y$ with $X \in \mathcal{F}$ via augmentations with single elements "inside $\mathcal{F}$". This is a strict relaxation of independence systems and can be thought of as an abstract generalization of connectivity in the sense that the family of all connected vertex sets of a graph always forms a strongly accessible set system. The main result in (Boley et al., 2007) is a DFS-algorithm that lists all closed sets of strongly accessible set systems with polynomial delay and incremental polynomial space. In this extended abstract we summarize our *new* results on listing closed sets of strongly accessible set systems:

(i) While listing all closed sets is intractable for accessible set systems, it becomes tractable for the class of strongly accessible set systems.

(ii) We give a *divide-and-conquer* algorithm listing all closed sets of strongly accessible set systems with polynomial delay and in polynomial space. This algorithm has not only *better complexity* than the DFS-algorithm in (Boley et al., 2007), but provides also an *algorithmic characterization* of strongly accessible set systems.

We investigate the relationship between closure operators and support-closedness of patterns with respect to datasets. Support-closedness is the closedness notion of mining transactional datasets: A set is called support-closed if all its supersets are contained in strictly less transactions than itself.

(iii) We show that support-closedness for all datasets is induced by a closure operator if and only if the set system satisfies a certain *confluence* property.

(iv) Moreover, a corresponding closure operator can be computed *efficiently* if its domain is strongly accessible. Together with the main result of this paper this constitutes a fairly general sufficiency criterion for the tractability of listing all support-closed patterns of a dataset.

(v) In contrast, if there is no corresponding closure operator, listing all support-closed sets is *hard* even for independence systems.

2. Preliminaries

A (finite) *set system* is an ordered pair $(E, \mathcal{F})$, where E is some (finite) set, called *ground set*, and $\mathcal{F} \subseteq \mathcal{P}(E)$. In this paper we consider only finite non-empty set systems. Furthermore, we assume that set systems are given implicitly by *membership oracles*. A membership oracle for $\mathcal{F}$ is a boolean-valued function that, for every $F \subseteq E$, returns "true" if and only if $F \in \mathcal{F}$.

A set system $(E, \mathcal{F})$ is called (i) *accessible* if for all $X \in \mathcal{F} \setminus \{\emptyset\}$ there is an $e \in X$ such that $X \setminus \{e\} \in \mathcal{F}$, (ii) *strongly accessible* if it is accessible and for all $X, Y \in \mathcal{F}$ with $X \subset Y$, there is an $e \in Y \setminus X$ such that $X \cup \{e\} \in \mathcal{F}$, and (iii) an *independence system* if $Y \in \mathcal{F}$ and $X \subseteq Y$ together imply $X \in \mathcal{F}$. Clearly, the class of strongly accessible set systems properly contains the class of independence systems and is properly contained by the class of accessible set systems. We define a further class of set systems that does not stand in any containment relation with the above classes. A set system $(E, \mathcal{F})$ is called *confluent* if for all $I, X, Y \in \mathcal{F}$ with $\emptyset \neq I \subseteq X$ and $I \subseteq Y$ it holds that $X \cup Y \in \mathcal{F}$.

We now turn to closure operators. Let $(E, \mathcal{F})$ be a set system. A mapping $\sigma : \mathcal{F} \to \mathcal{F}$ is called a *closure operator* if (i) $X \subseteq \sigma(X)$ (*extensivity*), (ii) $X \subseteq Y \Rightarrow \sigma(X) \subseteq \sigma(Y)$ (*monotonicity*), and (iii) $\sigma(X) = \sigma(\sigma(X))$ (*idempotence*) hold for all $X, Y \in \mathcal{F}$. A set $F \in \mathcal{F}$ is called *closed* if it is a fixpoint of σ, i.e., if $\sigma(F) = F$. The family of closed elements of $\mathcal{F}$ is denoted by $\sigma(\mathcal{F})$, i.e., $\sigma(\mathcal{F}) = \{F \in \mathcal{F} \colon \sigma(F) = F\}$. Note that in comparison to other work on closed set enumeration, the domain of the closure operator is some subset of $\mathcal{P}(E)$, and not $\mathcal{P}(E)$. Thus, in general, σ does not induce a *closure system* on $\mathcal{F}$.

3. Closed Set Listing

In this section we deal with the following problem:

Problem 1 (LIST-CLOSED-SETS) Given *a set system* $(E, \mathcal{F})$ *with* $\emptyset \in \mathcal{F}$ *and a closure operator* $\sigma : \mathcal{F} \to \mathcal{F}$, list *the elements of* $\sigma(\mathcal{F})$.

Algorithm 1 Divide & Conquer Closed Set Listing

Input : finite set system $(E, \mathcal{F})$ with $\emptyset \in \mathcal{F}$ and closure operator σ on $\mathcal{F}$,
Output: family of closed sets $\sigma(\mathcal{F})$

MAIN:

1: **print** $\sigma(\emptyset)$
2: LIST $(\sigma(\emptyset), \emptyset)$

LIST(C, B):

1: choose an element $e \in E \setminus (C \cup B)$ satisfying $C \cup \{e\} \in \mathcal{F}$ if such an e exists; otherwise **return**
2: $C' \leftarrow \sigma(C \cup \{e\})$
3: **if** $C' \cap B = \emptyset$ **then**
4: **print** C'
5: LIST (C', B)
6: **end if**
7: LIST $(C, B \cup \{e\})$

Consider Algorithm 1. It is based on the divide-and-conquer paradigm and applies recursively the following principle: For the current closed set C, first list all closed supersets of C containing some augmentation element e and then all closed supersets of C not containing e. This is a well-known listing scheme (see, e.g., (Gély, 2005)). However, in contrast to other closed set listing algorithms, it is defined for any $\mathcal{F} \subseteq \mathcal{P}(E)$ with $\emptyset \in \mathcal{F}$. The following results hold for Algorithm 1.

Theorem 2 *Let $(E, \mathcal{F})$ be a set system with $\emptyset \in \mathcal{F}$. Algorithm 1 lists $\sigma(\mathcal{F})$ exactly and non-redundantly for all closure operators σ on $\mathcal{F}$ iff $(E, \mathcal{F})$ is strongly accessible.*

Remark 3 *As a byproduct of Theorem 2 we get an* algorithmic characterization *of strongly accessible set systems for the identity map on $\mathcal{F}$ as closure operator: Let $(E, \mathcal{F})$ be a set system with $\emptyset \in \mathcal{F}$. Then Algorithm 1 for input $(E, \mathcal{F})$ given by a membership oracle and for the identity operator on $\mathcal{F}$ lists $\mathcal{F}$ exactly and non-redundantly iff $(E, \mathcal{F})$ is strongly accessible.*

We now turn to efficiency. Let $n = |E|$. Since $|\sigma(\mathcal{F})|$ can in general be as large as 2^n, there is no algorithm solving LIST-CLOSED-SETS in time polynomial in n. Thus, we aim for a good time bound *per closed set* and a good bound on the *delay*, i.e., the number of steps between the output of two successive sets. In addition the complexity of Algorithm 1 also depends on the representation of the set system and on the closure operator. Accordingly, we will study the time and space complexity of Algorithm 1 also in terms of those of (i) finding an augmentation element (line 1) and (ii) computing the closure of an element in $\mathcal{F}$ (line 2). We denote by T_a, S_a, T_σ, and S_σ the maximum time and space requirements of these operations for an input of size n, respectively. Note that the *augmentation problem* in line 1 can always be solved with $|E \setminus (C \cup B)| \leq n$ membership-queries. We still make T_a an explicit parameter of our results, because usually the problem can be implemented more efficiently than by this naïve approach.

Theorem 4 *Restricted to strongly accessible set systems, Algorithm 1 solves* LIST-CLOSED-SETS *with total time $O(|E|\ (T_a + T_\sigma)\ |\sigma(\mathcal{F})|)$, delay $O(|E|^2\ (T_a + T_\sigma))$, and space $O(|E| + S_a + S_\sigma)$.*

While Algorithm 1 is only correct for strongly accessible set systems, there might be other efficient algorithms that solve LIST-CLOSED-SETS for a broader class of set systems. Clearly, for any set system $(E, \mathcal{F})$ and closure operator σ on $\mathcal{F}$, $\sigma(\mathcal{F})$ can be listed in total time $O(2^n)$ by a deterministic algorithm that has ac-

cess to $\mathcal{F}$ only by means of membership oracle and closure computations, if the invocation of the membership oracle and the closure computation are both charged by unit time. Theorem 5 below not only shows that this bound cannot be substantially improved for accessible set systems, but also implies that there is no deterministic algorithm solving LIST-CLOSED-SETS for this problem fragment in output polynomial time, i.e., by an algorithm having a time complexity that is polynomially bounded in $n + |\sigma(\mathcal{F})|$.

Theorem 5 *For accessible set systems $(E, \mathcal{F})$ and closure operator σ on $\mathcal{F}$ such that $|\sigma(\mathcal{F})| \leq 2$, there is no deterministic algorithm that has access to $\mathcal{F}$ only by means of membership oracle and closure computations, and correctly solves problem* LIST-CLOSED-SETS *by invoking the membership oracle and computing the closure operator at most $2^{n/4}$ times where $n = |E|$.*

4. Support-Closed Sets

So far we have defined a closed set as a fixpoint of some closure operator. In data mining a different notion of closedness is used. To define it, we first recall some definitions from frequent pattern mining. A *dataset* over a set E is a multiset $\mathcal{D}$ of subsets of E. The elements of $\mathcal{D}$ are called *transactions*. We say that $\mathcal{D}$ is *non-redundant* if for all $e \in E$ there is a $D \in \mathcal{D}$ with $e \notin D$. For a set $X \subseteq E$, the *support set* of X w.r.t. $\mathcal{D}$, denoted $\mathcal{D}[X]$, is the multiset of transactions of $\mathcal{D}$ containing X. Based on support sets one can define the following notion of closedness: A set $X \in \mathcal{F}$ is *support-closed* if $X \subset Y$ implies $\mathcal{D}[X] \supset \mathcal{D}[Y]$ for every $Y \in \mathcal{F}$. By $\mathcal{SC}(\mathcal{F}, \mathcal{D})$ we denote the family of all support-closed sets in $\mathcal{F}$ w.r.t. $\mathcal{D}$. Note that in case $\emptyset \in \mathcal{F}$ it holds that a dataset $\mathcal{D}$ is non-redundant iff $\emptyset \in \mathcal{SC}(\mathcal{F}, \mathcal{D})$. We include this requirement in our problem definition for practical and technical reasons.

Problem 6 (LIST-SC-SETS**)** Given *a set system $(E, \mathcal{F})$ and a non-redundant dataset $\mathcal{D}$ over E,* list *the family of support-closed sets $\mathcal{SC}(\mathcal{F}, \mathcal{D})$.*

The two notions of closedness, based on support sets and based on closure operators, are not equivalent: there are set systems and datasets such that no closure operator exists having exactly the support-closed sets as fixpoints. Hence Algorithm 1 is not generally applicable to Problem 6. Indeed, even when restricted to independence systems, LIST-SC-SETS is intractable.

Theorem 7 *There is no algorithm solving* LIST-SC-SETS *restricted to independence systems in output polynomial time (unless **P=NP**).*

If, however, such a closure operator exists, we call it *support closure operator* of $\mathcal{F}$ w.r.t. $\mathcal{D}$. For the existence of the support closure operator for arbitrary non-redundant datasets we have the following characterization result.

Lemma 8 *Let $(E, \mathcal{F})$ be a set system. The support closure operator for $\mathcal{F}$ w.r.t. $\mathcal{D}$ exists for all non-redundant datasets $\mathcal{D}$ over E iff $(E, \mathcal{F})$ is confluent.*

Lemma 8 can be used to characterize the instances of LIST-SC-SETS that are also instances of LIST-CLOSED-SETS. But even in case that the support closure operator exists, it is unclear whether its computation is tractable. The following lemma states that if a support closure operator has a strongly accessible domain, it can be computed efficiently by reducing it to the augmentation problem (line 1 of Algorithm 1) for which again we denote the required time and space by T_a and S_a, respectively.

Lemma 9 *Let $(E, \mathcal{F})$ be a strongly accessible set system, and $\mathcal{D}$ a dataset over E. If the support closure operator of $\mathcal{F}$ w.r.t. $\mathcal{D}$ exists it can be computed in time $O(|E|\ (|\mathcal{D}| + T_a))$ and space S_a.*

Combining Theorem 4 with the results of this section, we can identify a fairly general, tractable subproblem of LIST-SC-SETS. While the theorem below may not yield the strictest bounds for concrete problems where more structural assumptions hold, its conditions can usually be checked easily and it serves as a baseline for more specialized methods.

Theorem 10 *Restricted to set systems that are confluent and strongly accessible* LIST-SC-SETS *can be solved with total time $O(|E|^2\ (|\mathcal{D}| + T_a)\ |\mathcal{SC}(\mathcal{F}, \mathcal{D})|)$, delay $O(|E|^3\ (|\mathcal{D}| + T_a))$, and space $O(|E| + S_a)$.*

Note that it is crucial for Theorem 10 that Theorem 4 holds for closure operators that are only a partial function of the power set of the ground set. The support closure operator is in general not defined for arbitrary members of the power set.

References

Boley, M., Horváth, T., Poigné, A., & Wrobel, S. (2007). Efficient closed pattern mining in strongly accessible set systems. *Proceedings of PKDD*, LNAI 4702, (pp. 382–389). Springer, Heidelberg.

Gély, A. (2005). A generic algorithm for generating closed sets of a binary relation. *Proceedings of ICFCA*, LNCS 3403, (pp. 223–234). Springer, Heidelberg.