Pairs Covered by a Sequence of Sets

Peter Damaschke

Department of Computer Science and Engineering
Chalmers University, 41296 Göteborg, Sweden
ptr@chalmers.se

Abstract. Enumerating minimal new combinations of elements in a sequence of sets is interesting, e.g., for novelty detection in a stream of texts. The sets are the bags of words occuring in the texts. We focus on new pairs of elements as they are abundant. By simple data structures we can enumerate them in quadratic time, in the size of the sets, but large intersections with earlier sets rule out all pairs therein in linear time. The challenge is to use this observation efficiently. We give a greedy heuristic based on the twin graph, a succinct description of the pairs covered by a set family, and on finding good candidate sets by random sampling. The heuristic is motivated and supported by several related complexity results: sample size estimates, hardness of maximal coverage of pairs, and approximation guarantees when a few sets cover almost all pairs.

1 Introduction

1.1 Motivation and Aim

In a chronological sequence of texts about some topic, such as a news stream, posts in social media, a timeline, etc., we may want to quickly understand what is novel in each entry, or what caused peaks in the volume of news about a topic. A simple approach is to determine new combinations of words. Ignoring the order of words, grammar, etc., let us consider the data as a sequence of sets ("bags of words"). The given bags may already be preprocessed: ignoring stop words, stemming, identifying synonyms, etc.

Definition 1. *Let $B_0, B_1, B_2, \dots, B_{m-1}$ be a sequence of sets that we call* bags. *For another bag $B := B_m$ we call a subset $X \subseteq B$* new *at m, if X was not already subset of an earlier bag: $\forall i < m : X \setminus B_i \neq \emptyset$. Otherwise $X \subseteq B$ is said to be* old *at m. We call $X \subseteq B$* minimal new *at m, if X is new and also minimal (with respect to inclusion) with this property.*

For knowing *all* new sets it suffices to know the *minimal* new sets. In the case of word sets, X can be a single new name or term, or a new pair, triple, etc., of old words, indicating new connections. They often give a good intuitive description of what is novel. Others can be understood only in context, or they are unrelated and meet only by chance. But before judging the new sets semantically one needs to find them first. Examples as below suggest that minimal new pairs are abundant, and minimal new sets of $h > 2$ words are rare. This is expected since X can be minimal new only if all its subsets appeared earlier.

Example. In a timeline of major discoveries in physics we should find an article about the first formulation of the law of conservation of energy. Then {conservation, energy} is minimal new, as this combination is new, but the term "energy" was coined earlier, and other conservation laws were formulated earlier, e.g., conservation of matter. Other articles will deal with the prediction and confirmation of electromagnetic waves. The pair {electromagentic, wave} is minimal new: They were totally unknown before, but physics had already dealt with other waves (such as light waves, being unaware that they are electromagnetic, too), and with other electromagnetic phenomena like induction.

Novelty mining and novelty detection in streams is extensively studied [2, 11, 12]. The subject is also related to minimal infrequent itemsets and mining emerging patterns [1, 8, 7]. However, in the present work we do not apply any language processing or machine learning to extract news or to produce online summaries, rather we explore the complexity of a combinatorial approach that relies on a very simple idea but gives meaningful hints to novelty. We consider the following problem, applied to sequences of texts like short articles.

Problem. Given a sequence of bags $B_0, B_1, B_2, \ldots$, enumerate the minimal new subsets X in each bag.

X is minimal new at m if and only if X is a minimal hitting set in the family of sets $\{B \setminus B_i \,|\, i < m\}$, that we call *hyperedges*. A set family is also called a *hypergraph*, and a *hitting set* (or *transversal*) intersects all hyperedges.

Define $n := \max |B_i|$, let m be the number of hyperedges (bags), and let c denote the number of minimal hitting sets (new sets at index m). Note that c is known only in hindsight, nevertheless one can express time bounds in terms of c. We may also fix a small number h and enumerate only hitting sets of size at most h. In particular, we focus on $h = 2$ due to the above motivation.

Algorithms for several parameterizations of minimal transversal enumeration are given in [4]: One can enumerate them with $O(n^2 m^2 e^{m/e})$ delay, hence in $O(n^2 m^2 e^{m/e} c)$ time, or alternatively in $O(n^2 c^3 e^{c/e} m)$ time. Actually one can enumerate any number $j \leq c$ of minimal hitting sets in $O(n^2 j^3 e^{j/e} m)$ time.)

A time bound is also provided in [4] for the case when the elements have *complementary degree* at most $q \ll m$, that is, every element appears in all but at most q hyperedges. (However, their result addresses only the verification of a given enumeration of the minimal transversals, not the construction.) In our application, the assumption means that words appear in at most q bags B_i, with $q \ll m$. This is sensible, because more frequent words are common words or stop words that are not informative and may be ignored.

1.2 Overview of Contributions

First we fill the mentioned gap in the parameterized complexity of transversal enumeration, but then we focus on small transversal sizes h, due to our motivation. The minimal new subsets of size h in each bag of size n in a sequence can be enumerated in $O(n^h)$ time. The rest of the work deals with the most relevant case $h = 2$, that is, new pairs. Note that we do not improve upon the $O(n^2)$ worst-case time per bag, but we study heuristics to save time for special structures that are however likely to appear in streams of topic-wise related texts: Large overlaps with earlier bags allow to recognize old pairs in (ideally) linear time. Saving a factor up to n in the processing time is worthwhile. Twin graphs give a succinct description of the pairs covered by a family of bags. We use them within a simple greedy heuristic to cover many old pairs. The greedy approach has known performance guarantees. We combine it with random sampling to find good candidate bags fast enough. Several complexity results justify this approach: We derive a logarithmic upper bound on the number of elements to be sampled in order to find bags with nearly largest overlaps with a given bag. We show that covering the exact maximum number of pairs by a prescribed number r of bags is $W[1]$-hard. We also propose to build larger bags from the given ones and show that some ternary set operation is sufficient for that. A final technical section is devoted to approximation guarantees for the number of covered pairs, if a few bags cover almost all pairs. We give a combinatorial approach to prove such results, based on special minimal set families and symmetries in convex optimization problems. Much of the latter parts is work in progress. We point to several open problems, highlighted as "Research question". The practical scenario might also be turned suitably into online problems. Besides the complexity aspects it would be interesting to test the approach on extensive data sets. – Due to lack of space, several proofs and proof details are omitted in this version.

2 Finding Minimal New Sets

2.1 Transversals for Small Complementary Degree

Theorem 1 below might result from [4] by reductions, however we present a self-contained algorithm description. Given a hitting set H and an element $s \in H$, we call E a *private hyperedge* if $s \in E$ and s is the only element of $H \cap E$. A hitting set H is minimal if and only if every $s \in H$ has at least one private hyperedge. We can assume that no hyperedge is subset of another one.

Theorem 1. *In hypergraphs $\mathcal{H}$ with complementary degree at most q we can enumerate all minimal hitting sets in $O(n^3 m q^2 e^{q/e} c)$ time.*

Proof. Loop through the $O(nm)$ pairs (s, E) of any element s and any hyperedge $E \ni s$ and do the following from scratch. Delete all hyperedges $F \ni s$ (they are hit by s) and all elements in E (in order to keep E private for s). At most q hyperedges remain, from which the elements of E are deleted. No hyperedge becomes empty, since no other hyperedge is a subset of E. In this "small" instance enumerate all minimal hitting sets H. Every $H \cup \{s\}$ is a minimal hitting set of $\mathcal{H}$: It intersects all hyperedges, every element has a private hyperedge since H was minimal, and s has the private hyperedge E. Conversely, all c minimal hitting sets of $\mathcal{H}$ are obtained in this way. To avoid duplicates, first sort the elements arbitrarily, and demand the above s to be the first element in the hitting sets, that is, delete also elements before s. Now some pairs (s, E) yield no solution since some hyperedges lose all their elements, but we recognize these cases instantly. The "small" instances are solved by the $O(n^2m^2e^{m/e})$-delay algorithm for hitting set enumeration from [4]; replace m with q. Some original edges may become identical due to deletions, which does not affect the time bound. Finally note that every "small" instance is prepared in $O(nm)$ time. □

2.2 Enumerating the Minimal New Sets in a Sequence

Unlike the previous parameterizations, for the text stream applications we need small transversal size h and time bounds that are polynomial in q and m and also have a better dependency on n. Naive exhaustive search takes every $X \subseteq B_m$, $|X| \leq h$, and checks in $O(hq)$ time whether X intersects all hyperedges $B_m \setminus B_i$. (For each element s we maintain the at most q indices with $s \in B_i$. An $s \in X$ misses at most q hyperedges, and we see whether the other elements of X hit them.) This way we would need $O(hqn^h/h!) = O(hq(en/h)^h) = O(n^h \cdot q \cdot h(e/h)^h)$ time to find all minimal new sets of size h in B_m. However we can avoid checking all $X \subseteq B_m$ against all previous $B_m \setminus B_i$ $(i < m)$: We generate the candidate sets X with increasing sizes and, due to minimality, stop as soon as all further X would be supersets of already detected minimal new sets at m. More importantly, we can use information about the minimal new sets at earlier $i < m$, as detailed below.

Let $f(X) := \min\{i | X \subseteq B_i\}$, and let $f(X)$ be undefined if no such i exists. Note that $f(X) = i$ if and only if X is new at i (but not necessarily minimal). Furthermore, X is then old at any further index $j > i$. In the following we assume for simplicity that a dictionary operation costs constant time per element. We store some sets X along with $f(X)$ in a dictionary. In particular, whenever a set X is minimal new at i, we store "$f(X) = i$". This is no extra work, since our aim is to enumerate all minimal new sets. So suppose that we have already determined all minimal new sets with at most h elements at all $i < m$. In particular, every single element is new as soon as it appears for the first time. Hence we can check every single element for being new, in $O(nm)$ time in total. After these precautions we get for $B = B_m$:

Theorem 2. *The minimal new subsets of size at most h in a bag of n elements can be enumerated in $O(n^h \cdot 2^h/h!)$ time.*

Proof. Suppose that we have already determined all minimal new sets with fewer than j elements. Consider any set $X \subseteq B_m$ of j old elements, such that no $Y \subset X$ is new at m. In order to check whether X is new at m, thus minimal new, we proceed as follows. If a value $f(X) < m$ is stored, then X is old. Suppose that no $f(X)$ value is stored. Then we must figure out whether $X \subseteq B_i$ for some $i < m$. Assume that such an index exists, and let i be the smallest one. Since no $f(X)$ value is in place, X is not minimal new at i, and this is possible only if some nonempty $Y \subset X$ is minimal new at i, in particular, $f(Y) = i$ has been stored. Thus we must only look up the $2^j - 2$ values $f(Y)$ and, if $f(Y)$ is stored, check whether $X \subseteq B_{f(Y)}$. If we find such Y, then we conclude that X is old at m, and we can also store $f(X)$ which equals the smallest such $f(Y)$. Otherwise we conclude that X is minimal new at m, and we also store $f(X) := m$. Altogether we can decide in $O(2^j)$ time whether a set X of size j is minimal new. The number of sets X to check is at most $\binom{n}{j} < n^j/j!$. The procedure is repeated for increasing j up to h, and $j = h$ dominates the time bound. □

3 A Heuristic for Minimal New Pairs

3.1 Below the Worst-Case Quadratic Time

The remainder of the paper deals with the case $h = 2$ only. As argued earlier, minimal new pairs, i.e., new pairs of old elements, seem to be very good indicators of novelty in text streams. (Also note that the case $h = 1$ is trivial.) Let $C_j := B_j \cap \bigcup_{i<j} B_i$ denote the set of old elements in B_j. We define the *index set* of an element $s \in C_j$ as $\{i|\ i < j,\ s \in B_i\}$. While watching the sequence we can collect the old elements and thus obtain the sets C_j in $O(\sum_j |B_j|)$ overall time, and maintain the index sets in $O(\sum_j |C_j|)$ overall time. After this trivial auxiliary processing, the problem for each bag can be stated as follows:

Problem. In a sequence of bags $B_0, B_1, B_2, \ldots$ enumerate the new pairs in $C := C_m = B_m \cap \bigcup_{j<m} B_j$, that is, pairs that are not covered by earlier B_j, $j < m$. Besides the B_j, the index sets of all elements in C are already given. Let $n := |C|$. We also refer to the $B_j \cap C$ as *bags*, without risk of confusion.

By Theorem 2 we can recognize the new pairs in C in $O(n^2)$ time in the worst case. The interesting matter is to enumerate them faster if only a minority of pairs in C is new: Note that, once we detect a large bag $B_j \cap C$, we can immediately exclude the pairs therein as candidates for new pairs, in $O(|B_j \cap C|)$ time rather than $O(|B_j \cap C|^2)$. Large intersections are likely in streams of related texts, as they tend to form topics clusters with similar word content.

Driving the idea one step further, we may select a number r of bags, exclude all pairs covered by them, and test only the uncovered pairs for being new, each in $O(1)$ time. However the total time for finding such bags and listing the uncovered pairs must be $o(n^2)$, therefore we will have to use, in general, some $r < m$ bags. In the following we elaborate on this idea.

3.2 The Twin Graph

Suppose that we have already selected r bags. The set of pairs (not) covered by them is completely described by the following structure.

Definition 2. *With respect to a set of r bags, any two elements of C with the same index set (i.e., elements being in exactly the same bags) are called* twins. *This yields an equivalence relation on C whose t equivalence classes are called* twin classes. *The* twin graph *is the graph whose vertices are the twin classes, and where any two vertices with disjoint index sets are joined by an edge.*

Some properties of the twin graph are obvious: It has a self-loop only at the vertex representing elements that are in no bag (if existing). The uncovered pairs of elements are exactly those in any two adjacent twin classes. The twin graph has $t \leq \min(n, 2^r)$ vertices and at most $\frac{1}{2}\min(t^2, 3^r)$ edges, since the r indices can be partitioned in $\frac{1}{2} \cdot 3^r$ ways in two disjoint index sets and the rest.

Proposition 1. *The twin graph of r bags can be constructed iteratively, that is, by inserting the bags one by one, in $O(r \cdot (\min(t^2, 3^r) + n))$ time.*

Proof. Every bag may split some twin classes in two smaller ones. These splittings are done in $O(n)$ time per bag, hence $O(rn)$ time overall. Index sets are stored as a tree in an obvious way. To obtain the edges we either check the $O(t^2)$ pairs of twin classes for disjointness of their index sets in $O(rt^2)$ time, or we take all $O(3^r)$ possible pairs of disjoint index sets and check their existence, which can be done in $O(r)$ time for each pair. □

3.3 Greedy Partial Set Cover of Pairs

We need to determine the bags to be inserted and to update their twin graph. We can stop as soon as inserting another bag, that covers p new pairs, requires $O(p)$ time, i.e., the time needed to simply test these pairs individually for being new, as in Theorem 2. As Proposition 1 indicates that the time to update the twin graph can grow exponentially in the number r of bags, let us fix a number r of bags (which may however depend on n, say, some $r = o(\log n)$), and aim at solving the following problem.

Partial Set Cover of Pairs. Given a family of bags and a number r, identify r bags that together cover the maximum number of pairs.

Proposition 2. Partial Set Cover of Pairs *is NP-complete, and also $W[1]$-complete in the parameter r.*

Proof. Reduction from Independent Set. Omitted due to space limits. □

Due to this negative observation we resort to a greedy approach: The Partial Set Cover problem asks to cover a maximum number of elements by a prescribed number r of bags. The greedy algorithm for (Partial) Set Cover iteratively adds to the solution a bag with the largest number of yet uncovered elements. The number of elements covered in r greedy steps is at least a $1-(1-1/b)^r$ fraction of the optimum that could be covered by b bags. This was shown in [5], generalizing an earlier result in [10, 9] for $r = b$. This bound is also tight for $r = b$ [10], and the worst-case example for $r = b$ also works in general.

Now let P be the set of the $\binom{n}{2}$ pairs of elements in C, and let P_j be similarly defined for each $B_j \cap C$. We refer to the P_j also as bags, without risk of confusion. Since the P_j and P are made of pairs of other elements, Partial Set Cover of Pairs is a special case of Partial Set Cover. Thus, the greedy algorithm gives at least the same approximation guarantee.

Research question. Figure out the approximation ratio for greedy Partial Set Cover of Pairs. We conjecture that it is significantly better than for Partial Set Cover. However, the case of pairs appears to be intrinsically more difficult. While the proof in [5] is merely based on the pigeonhole principle, we must also deal with the twin graph structure. (Section 5 will give a method to obtain some results for the case when r bags cover almost all pairs.)

4 Supplementary Results

4.1 Sampling Large Intersections

Since $o(n^2)$ time is mandatory, implementation of the greedy heuristic needs some care. Besides updating the twin graph we have to count in the time for finding the next bag that covers as many further pairs as possible. As long as m is small compared to n we can afford computing all intersections $B_j \cap C$ in $O(mn)$ time. For larger m we may sample some random elements or pairs from C, use their index sets (of size at most q, typically much smaller than m) to count their occurences in the given bags, and take the bags with most hits.

Theorem 3. *Suppose that the largest bag has $(1-x)n$ elements, and we sample s random elements and return the bag with the largest number of hits. Then we fail to find a bag with at least $(1-y)n$ elements with probability at most $2m \cdot \exp(-(y-x)^2 s/16y)$. In particular, we get a failure probability below any prescribed constant by choosing $s = \Theta(\log m \cdot y/(y-x)^2)$.*

Proof. By Chernoff bounds. Omitted due to space limits. □

In particular, we need $O((q \log m)/y)$ time to find a bag whose size is at least $1-y$ of the maximum size of a bag, where q denotes the maximum size of the index sets. (Fix x in Theorem 3 and note that we must traverse the index set of every sample.) While Theorem 3 was formulated for elements, it applies literally to sampling of pairs, too, and can be used in each step of the greedy heuristic: Some bags are already selected, and they form a twin graph with t vertices. Then the uncovered pairs form the edge sets of $O(t^2)$ cliques and bicliques of known sizes, thus one can easily sample random uncovered pairs from them.

4.2 Building Larger Bags

For a sequence of bags $B_0, B_1, B_2, \ldots, B_{m-1}$ consider the graph G whose vertices are the elements, where two vertices u, v are adjacent if and only if $u, v \in B_i$ for some i. A *clique edge cover* in a graph is a set of cliques that cover all edges. Hence the bags form a clique edge cover of G. However, G may contain further, larger cliques, and using them besides the given bags within our heuristic is beneficial, since they cover more of the old pairs. If we can quickly find and build some of these larger cliques, we can use them later in the sequence and make later steps more efficient. Bags with large intersections inside the current bag B_m (that are anyway used to cover many old pairs in B_m) are likely to have large intersections also outside B_m. We may apply the following ternary set operation Δ on them: $\Delta(X,Y,Z) := (X \cap Y) \cup (X \cap Z) \cup (Y \cap Z)$. Note that each pair of elements in $\Delta(X,Y,Z)$ is also contained in some of X,Y,Z. In particular, if X,Y,Z are any three cliques, then $\Delta(X,Y,Z)$ is a clique, too. A neat fact is that all maximal cliques can be generated from any clique edge cover using only the Δ operation. This supports the idea to apply Δ to bags that are anyhow considered in a step of the heuristic.

Proposition 3. *Given a graph along with a clique edge cover $\mathcal{K}$, we can obtain every maximal clique solely by repeated Δ operations applied to cliques of $\mathcal{K}$.*

Proof. By induction on the size. Omitted due to space limits. □

The sizes of bags produced by Δ can grow quickly, by a factor up to $\frac{3}{2}$ (if $X = Y' \cup Z'$, $Y = X' \cup Z'$, $Z = X' \cup Y'$ for three disjoint sets X', Y', Z' of equal size. On the combinatorial side it would be interesting to know what cliques size are guaranteed to exist in a graph with few non-edges compared to $\binom{n}{2}$. Turán's theorem [13] states, informally, that a graph with few non-edges always has a large clique. We would be interested in a generalization to unions of r cliques:

Research question. Given r, n, u, what is the guaranteed number of edges that can be covered by r cliques, in any graph of n vertices and $\binom{n}{2} - u$ edges? While some ideas from the proof of Turán's theorem generalize to a union of r cliques, the extremal problem apparently becomes harder.

We remark that another conceivable approach for obtaining larger bags would be to see if some r bags can be replaced with $k < r$ new bags that cover the same edges. This amounts to the Clique Edge Cover problem parameterized by k. This NP-hard problem is fixed-parameter tractable in parameter k [6], however with doubly exponential time bound, and non-existence of a polynomial kernel [3] leaves little hope to reduce this asymptotic worst-case bound.

5 Approximations if a Few Bags Cover All Pairs

5.1 Setup and Preparations

Proposition 2 raises the question: What fraction of pairs in C can we cover within some $O(m^{O(1)}n)$ time bound? Due to the context, we are interested in the case that r bags exist that cover all pairs in C, subject to some small fraction. Note that m may here denote the number of sampled candidate bags (having large intersections with C) rather than the length of the entire sequence of sets.

More specifically, suppose that some r bags cover $(1-\delta)\binom{n}{2}$ pairs, where $\delta > 0$ is a small number. Consider their twin graph (Definition 2), with t vertices. If two adjacent twin classes have $\sqrt{\delta}n$ elements each, then already δn^2 pairs of elements are uncovered, contradicting the assumption. Hence the twin graph has a vertex cover of twin classes with fewer than $\sqrt{\delta}n$ elements each. The complement of this vertex cover is an independent set in the twin graph, thus representing a subset C' of C with at least $(1-t\sqrt{\delta})n$ elements in which all pairs are covered. (Actually the fraction is closer to 1; the given bound is coarse only due to the general argument.) Thus we may clean up our question as follows:

Problem. In a family of m bags, suppose that r of them cover all pairs in C', for some $C' \subset C$ of size $n' := |C'| > (1-\epsilon)|C|$. (But note that C' is not specified in the input.) What number $\gamma\binom{n'}{2}$ of pairs can we at least cover within some $O(m^{o(r)}n)$ time bound? We call the fraction γ the *coverage*.

In the following we work with the complement of the twin graph restricted to C', which is a clique of some $t' \leq t$ vertices whose edges are covered by r smaller cliques (bags). We suppose that r is minimal, that is, $r-1$ of the bags would not cover all pairs. Let A be the *incidence matrix* of these bags: A has a row for every bag, a column for every twin class, and entries $a_{ij}=1$ if bag i contains the twin class j, and $a_{ij}=0$ otherwise. Hence every column is the characteristic vector of an index set. A pair of twin classes is *private* for a bag if that bag covers that pair but none of the other $r-1$ bags does. We establish some properties of A. Since r is minimal, no row is contained in another row, and since the bags cover all pairs, the columns pairwise intersect. In other words:

(1) For any two rows i and i' there exists some columns j and j' such that $a_{ij}=a_{i'j'}=1$ and $a_{ij'}=a_{i'j}=0$.

(2) For any two columns j and j' there exists some row i with $a_{ij}=a_{ij'}=1$.

Since we are only interested in worst-case approximation ratios, we can assume further restrictions: Deletion of any twin class from any bag must destroy some private pair, since otherwise there would exist a worse instance with smaller bags, such that the coverage can only decrease. In other words, every twin class in a bag belongs to some private pair of that bag. More formally:

(3) For each entry $a_{ij}=1$ there exists a column j' such that $a_{ij'}=1$, and $a_{i'j}=0$ or $a_{i'j'}=0$ holds for each row $i' \neq i$.

We remark that (3) implies (1). Another conclusion is that any two index sets are incomparable, that is, not in subset relation:

(4) For any two columns j and j' there exists some rows i and i' such that $a_{ij}=a_{i'j'}=1$ and $a_{ij'}=a_{i'j}=0$.

To show (4), let i'' be some row according to (2): $a_{i''j}=a_{i''j'}=1$. There must be some row i where $a_{ij}\neq a_{ij'}$, since equal columns would represent the same twin class. Suppose $a_{ij}=1$ and $a_{ij'}=0$. Due to (3), there exists a column j'' such that $a_{ij''}=1$. Since the pair represented by $a_{ij}=a_{ij''}=1$ is private and $a_{i''j}=1$, it also follows $a_{i''j''}=0$. Condition (2) applied to j' and j'' yields the existence of another row i' with $a_{i'j'}=a_{i'j''}=1$. Finally, since the pair represented by $a_{ij}=a_{ij''}=1$ is private and $a_{i'j''}=1$, it follows $a_{i'j}=0$.

The following consideration of symmetries will help reduce case distinctions. Automorphisms of an optimization problem are permutations of the variables that leave the set of constraints invariant. An orbit of the automorphism group is a set of variables mapped onto each other by automorphisms; clearly they form equivalence classes. For convex minimization problems it is known that the convex combination of any two minimal solutions is a minimal solution, too. From this it follows: If we take any minimal solution, apply all automorphisms to the variables, and take component-wise the average of all these solutions, we obtain a minimal solution where all variables in each orbit have equal values.

Consider an optimization problem with variables $x_1, \ldots, x_t$ and y, with the objective $\min y$, and constraints $g_j(x_1, \ldots, x_t) \le y$ and $h_j(x_1, \ldots, x_t) \le 0$, where all g_j and h_j are convex functions. Such a problem is convex and can be rephrased as $\min \max_j g_j(x_1, \ldots, x_t)$ under the constraints $h_j(x_1, \ldots, x_t) \le 0$.

5.2 Illustration: Some Approximation Guarantees

Now we apply these tools to determine set families that minimize certain guaranteed approximation ratios. Since only $o(n^2)$ time bounds matter, the smallest r are most relevant for us. To avoid technicalities we assume large enough n such that we can neglect lower-order terms and the effects of rounding fractional numbers to integers. The case of $r \le 2$ bags is trivial. Let $O(m^\omega)$ be a time bound for multiplying $m \times m$ matrices.

Theorem 4. *If r of $m < n$ bags cover all pairs in C, we find r such bags in $O(m^{\omega-1}n) = o(m^2 n)$ time if $r \le 4$, and in $O(m^{r-2}n + (2m)^r)$ time if $r > 4$.*

Proof. Let A again denote the $0,1$-incidence matrix whose rows represent bags. First we exclude in $O(mn)$ time the trivial case that some row has only 1s. We call a set R of rows unsuitable if R has some all-0 column, and suitable otherwise. Let J be the graph whose m vertices are the rows, with an edge in every suitable pair. We compute J by switching 0s and 1s in A and multiplying this matrix with its transpose in $O((n/m)m^\omega = O(m^{\omega-1}n)$ time. Let R denote a set of r rows (bags) or the corresponding $r \times n$ submatrix of A. For $r = 3$ we have: R covers all pairs if and only if any pair of rows in R is suitable. For $r \ge 4$ we have: R covers all pairs if and only if R is not the union of two unsuitable sets S, T with $|S| + |T| = r$ and $2 \le |S| \le |T| \le r - 2$. (Recall that the trivial case was ruled out.) For $r = 3$ we need to find a triangle in J, which is well known to work in time $O(m^\omega) = O(m^{\omega-1}n)$, as $m < n$ was assumed. For $r = 4$ we also need to find a triangle plus any fourth vertex, or a star of three edges in J. The latter is trivially done in $O(m^2) = O(mn)$ time. For $r > 4$ we determine all unsuitable sets of at most $r - 2$ rows in $O(m^{r-2}n)$ time. The condition for each R is then checked in $O(2^r)$ time. □

A slight relaxation of this algorithm handles the case when r bags cover all pairs in C': Then we call a pair of rows suitable when at most ϵn columns lack a 1, apply the arguments to C', and find r bags with coverage $1 - \Theta(\epsilon)$ in the same time. – The following proofs show the existence of bags with certain minimum numbers of elements in C', implying coverage values if we would choose these bags. Actually we select bags of that size in C rather than in the unknown $C' \subset C$, which can only increase the coverage due to smaller overlaps.

Proposition 4. *If three out of m bags cover all pairs in C', then we can find one bag with coverage $4/9$ in $O(mn)$ time, and two bags with coverage $7/9$ exist.*

Proof. We remark that the pigeonhole principle would only yield coverage $1/3$ and $2/3$, respectively. Any three bags satisfying conditions (1)–(4) have the form $T_1 \cup T_2$, $T_1 \cup T_3$, $T_2 \cup T_3$, with twin classes T_1, T_2, T_3. From the m given bags we take one bag, and a pair of bags, respectively, with the largest coverage. Let $x_i := |T_i|$. For a lower bound on the coverage we minimize the maximum number of pairs covered by one or two of the considered bags. The numbers of covered pairs are convex functions of the x_i, the only orbit is $\{x_1, x_2, x_3\}$, and the constraints $x_i \geq 0$ and $x_1 + x_2 + x_3 = n'$ are linear. Now the above symmetry consideration yields that $x_1 = x_2 = x_3 = \frac{1}{3}$ is the minimal solution in both cases, and finally, simple calculations yield the ratios. □

Two bags with coverage $7/9$ could be found in $O(m^2 n)$ time, but Theorem 4 achieves already more. We mention the coverage $7/9$ only as a benchmark for the next result that sacrifices some coverage for speed.

Proposition 5. *If three out of m bags cover all pairs in C', then we can find two bags with coverage $56/81 > 0.69$ in $O(mn)$ time.*

Proof. A greedy strategy first takes a largest bag G and then adds a second bag that covers a maximum number of further pairs. Three bags that minimally cover all pairs have a structure as in Proposition 4, with each twin class further split in two by G. This results in six twin classes $T_1, \ldots, T_6$ such that $G = T_1 \cup T_2 \cup T_3$, and $T_1 \cup T_2 \cup T_4 \cup T_5$, $T_2 \cup T_3 \cup T_5 \cup T_6$, $T_1 \cup T_3 \cup T_4 \cup T_6$ are mentioned three bags. Note that $x_1 + x_2 + x_3 \geq \frac{2}{3} n'$, and the second greedy step can take some of these three bags (or a better one). For a lower bound on the coverage we minimize the maximum number of pairs covered by these three choices. The numbers of covered pairs are convex functions of the x_i, the orbits are $\{x_1, x_2, x_3\}$ and $\{x_4, x_5, x_6\}$, and the constraints $x_i \geq 0$, $x_1 + x_2 + x_3 \geq \frac{2}{3} n'$ and $x_1 + \cdots + x_6 = n'$ are linear. Symmetry consideration yields the existence of a minimal solution where $x_1 = x_2 = x_3$ and $x_4 = x_5 = x_6$. Moreover, $x_1 + x_2 + x_3 = \frac{2}{3} n'$ is the worst case, and simple calculations yield the ratios. □

Proposition 6. *If four out of m bags cover all pairs in C', then we can find one bag with coverage $9/25$ in $O(mn)$ time.*

Proof. By case inspections, any four bags satisfying the conditions (1)–(4) have the form $T_1 \cup T_2 \cup T_3$, $T_1 \cup T_4$, $T_2 \cup T_4$, $T_3 \cup T_4$, with twin classes T_1, T_2, T_3, T_4, and then similar arguments as above apply, with orbits $\{x_1, x_2, x_3\}$ and $\{x_4\}$. □

Research question. Systematically explore the trade-off between time and coverage for any r and number s of greedy bags. The numbers of pairs covered by bags are sums of squares and products of the induced twin class sizes, thus always convex. But with growing r and s the task becomes more challenging, as we need to understand the combinatorics of the "minimal" coverings and their binary incidence matrices that obey conditions (1)–(4). It might be possible to combine the pigeonhole principle with our stronger method. Moreover, does the relationship to matrix multiplication also yield lower time bounds (see [14])?

Acknowledgment

This work has been supported by the Swedish Foundation for Strategic Research (SSF) through Grant IIS11-0089 for a data mining project entitled “Data-driven secure business intelligence”. The author also wishes to thank the referees for careful reading.

References

1. Boros, E., Gurvich, V., Khachiyan, L., Makino, K.: On Maximal Frequent and Minimal Infrequent Sets in Binary Matrices. Ann. Math. Artif. Intell. 39, 211–221 (2003)
2. Ceci, M., Appice, A., Loglisci, C., Caruso, C., Fumarola, F., Valente, C., Malerba, D.: Relational Frequent Patterns Mining for Novelty Detection from Data Streams. In: Perner, P. (Ed.) MLDM 2009. LNCS, vol. 5632, pp. 427–439, Springer, Heidelberg (2009)
3. Cygan, M., Kratsch, S., Pilipczuk, M., Pilipczuk, M., Wahlström, W.: Clique Cover and Graph Separation: New Incompressibility Results. ACM Trans. Comput. Theory 6 (2014), Article 6
4. Elbassioni, K.M., Hagen, M., Rauf, I.: Some Fixed-Parameter Tractable Classes of Hypergraph Duality and Related Problems. In: Grohe, M., Niedermeier, R. (Eds.) IWPEC 2008. LNCS, vol. 5018, pp. 91–102, Springer, Heidelberg (2008)
5. Elomaa, T., Kujala, J.: Covering Analysis of the Greedy Algorithm for Partial Cover. In: Elomaa, T., Mannila, H., Orponen, P. (Eds.): Algorithms and Applications. Essays Dedicated to Esko Ukkonen on the Occasion of His 60th Birthday, LNCS 6060, pp. 102–113, Springer, Heidelberg (2010)
6. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Data Reduction and Exact Algorithms for Clique Cover. ACM J. Exper. Algor. 13 (2008)
7. Gupta, A., Mittal, A., Bhattacharya, A.: Minimally Infrequent Itemset Mining using Pattern-Growth Paradigm and Residual Trees. In: Haritsa, J.R., Dayal, U., Deshpande, P.M., Sadaphal, V.P. (Eds.) 17th Int. Conf. on Management of Data, pp. 57–68, Allied Publishers, Bangalore (2011)
8. Haglin, D.J., Manning, A.M.: On Minimal Infrequent Itemset Mining. In: Stahlbock, R., Crone, S.F., Lessmann, S. (Eds.) DMIN 2007, pp. 141–147, CSREA Press 2007
9. Hochbaum, D.S.: Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems. In: Hochbaum, D.S. (Ed.) Approximation Algorithms for NP-hard Problems, pp. 94–143. PSW Publishing, Boston (1997)
10. Hochbaum, D.S., Pathria, A.: Analysis of the Greedy Approach of Maximum k-Coverage. Naval Research Quarterly 45, 615–627 (1998)
11. Karkali, M., Rousseau, F., Ntoulas, A., Vazirgiannis, M.: Efficient Online Novelty Detection in News Streams. In: Lin, X., Manolopoulos, Y., Srivastava, D., Huang, G. (Eds.) WISE 2013. LNCS, vol. 8180, pp. 51–71, Springer, Heidelberg (2013)
12. Karkali, M., Rousseau, F., Ntoulas, A., Vazirgiannis, M.: Using Temporal IDF for Efficient Novelty Detection in Text Streams. CoRR abs/1401.1456 (2014)
13. Turán, P.: ”On an Extremal Problem in Graph Theory. Matematikai és Fizikai Lapok 48, 436–452 (1941)
14. Williams, V.V., Williams, R.: Subcubic Equivalences Between Path, Matrix and Triangle Problems. In: FOCS 2010, pp. 645–654, IEEE Computer Society (2010)

Appendix

Proof of Proposition 2

We give a reduction from the NP-complete and $W[1]$-complete Independent Set problem. Let $G = (V, E)$ be any graph. For every vertex $v \in V$ we create a set K_v. Initially, every K_v consists of only one element, and these elements for all $v \in V$ are distinct. For every edge $vw \in E$ we add two fresh elements to both K_v and K_w, note that $K_v \cap K_w$ now contains one pair. Eventually we add further distinct elements to all K_v such that they all get equal sizes. A set of r bags of equal sizes covers the maximum possible number of pairs if and only if all these bags are disjoint, that is, if the corresponding vertices of G form an independent set. This establishes equivalence.

As an additional remark, this reduction constructs instances with a small coverage of pairs, however we can add to every K_w some large common set of further elements. Hence the problem remains $W[1]$-complete also in instances interesting for us, where r bags cover a large fraction of all pairs.

Proof of Theorem 3

An undesired event is that a larger bag gives a smaller count. First consider any two bags $B_i \cap C$ and $B_j \cap C$, and define x and y by $|(B_i \setminus B_j) \cap C| = xn$ and $|(B_j \setminus B_i) \cap C| = yn$, where $x < y$. We sample s random elements from C with repetition. Hence the expected number of hits of both set differences is together $(x + y)s$. By a Chernoff bound, the probability to hit the set differences fewer than $(1 - a)(x + y)s$ times is at most $\exp(-a^2(x + y)s/2)$.

We consider the event that the smaller set difference is hit more often than the larger one, first on the condition that we hit them together exactly $(1-a)(x+y)s$ times. This conditional event happens if and only if we hit the larger set difference at most $(1-a)(x+y)s/2 = (1-(y-x)/2y)(1-a)ys$ times while the expectation is $(1 - a)ys$. Using a Chernoff bound again, the probability for that is at most $\exp(-(y - x)^2(x + y)(1 - a)s/8y^2)$. If we hit the set differences more often, the same calculation applies with the same x, y, a and increased s, hence the upper probability bound remains valid. Thus, by the law of total probability, the bound remains valid also conditional on hitting the set differences *at least* $(1-a)(x+y)s$ times. By the union bound, the probability that the larger bag yields the smaller count is at most $\exp(-a^2(x+y)s/2)+\exp(-(y-x)^2(x+y)(1-a)s/8y^2)$. Choosing $a = (y - x)/2y$ and hence $1 - a = (y + x)/2y$, this becomes

$$\exp(-2y(x + y)(y - x)^2 s/16y^3) + \exp(-(x + y)^2(y - x)^2 s/16y^3)$$

which is finally bounded by $2\exp(-(x+y)^2(y-x)^2 s/16y^3)$. This can be further limited by the simpler $2\exp(-(y-x)^2 s/16y)$. By comparing the largest bag to all others and applying the union bound we obtain the claimed failure probability.

Proof of Proposition 3

We rephrase the assertion as follows: Every clique K in the graph is subset of a clique that can be obtained from $\mathcal{K}$ by repeated Δ operations. In this form we can prove it by induction on $i = |K|$. Induction base $i = 2$ is obviously true, since $\mathcal{K}$ is a clique edge cover. Assume that the assertion holds for all sizes from 2 to i, and consider a clique K of size $i+1$. Fix any vertices $u, v \in K$ and define $K_u := K \setminus \{u\}$ and $K_v := K \setminus \{v\}$. The induction step is now established by observing $K = \Delta(K_u, K_v, \{u, v\})$.

Details Omitted in Section 5.2

We consider the matrices whose rows and columns represent bags and twin classes, respectively, that satisfy conditions (1)–(4). We do not distinguish between a row (or column) and the set of its 1 entries.

The matrix must contain the submatrix

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

where the 1s in row 1 form a private pair. Rows 2 and 3 exist since no column contains another one. The submatrix extends to:

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & ? \end{pmatrix}$$

Namely, since row 1 does not contain row 2, there must be another 1 in row 2, and we can choose it such that the 1s in row 2 form a private pair. Therefore we get a 0 in row 1.

If $r = 3$ then the question mark must be 1, since columns 2 and 3 must intersect.

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Another column cannot exist, as it would be contained in some of these.

The fractions of covered pairs are convex[1] functions: $(x_1 + x_2)^2$ for one bag, $(x_1+x_2)^2+(x_2+x_3)^2-x_2^2 = x_1^2+x_2^2+x_3^2+2x_1x_2+2x_2x_3$ for two bags, and all symmetric cases. Since the maximum is minimized if $x_1 = x_2 = x_3 = 1/3$, each bag has $2/3$ of the elements. Thus, one bag has coverage $(2/3)^2 = 4/9$, and two bags that intersect in one twin class have coverage $2(2/3)^2 - (1/3)^2 = 7/9$.

The fraction of pairs covered by two greedy bags has the same structure (sum of two squares minus a smaller square) and is therefore a convex function, too,

[1] Convexity of functions can be checked with the help of their Hesse matrices.

and the numerical value becomes $2(2/3)^2 - (4/9)^2 = (72-16)/81 = 56/81$, as the intersection now contains $4/9$ of the elements.

We turn to $r = 4$. If all columns of the matrix have only two 1s, the only matrices whose columns also pairwise intersect (up to permutations of rows and columns, of course) are

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and the previous matrix with an all-0 row attached. But in both cases some rows contain others. Similarly, an all-1 column would contain all others. Hence some column with exactly three 1s must exist. Suppose we have two such columns, like column 1 and 2 in this matrix:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & ? & ? \\ 0 & 1 & ? & ? \end{pmatrix}$$

Columns 3 and 4 are enforced by the condition that neither of row 1 and 2 can contain the other one. Next, since neither of columns 3 and 4 is contained in column 1 or 2 intersect, all wildcards are necessarily 1:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

But now, none of the pairs of 1s in any row is private, which enforces more columns, but any further column would be contained in some of the displayed columns. This contradiction shows that exactly one column has exactly three 1s, hence the only remaining possibility is:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Let $x := x_1 = x_2 = x_3$, thus $x_4 = 1 - 3x$. Note that the bags have sizes $3x$ (row 4) and $x+(1-3x) = 1-2x$ (else). The largest bag has $\max\{3x, 1-2x\}$ of the elements, which is minimized for $x = 1/5$, hence the coverage is $(3/5)^2 = 9/25$. The pigeonhole principle would only yield $1/4$.