Relative Risk and Odds Ratio: A Data Mining Perspective (Corrected Version)

Haiquan Li, Jinyan Li, & Limsoon Wong[*]
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{haiquan, jinyan, limsoon}@i2r.a-star.edu.sg

Mengling Feng & Yap-Peng Tan
Nanyang Technological University
Block S1, 50 Nanyang Ave, Singapore 639798
{feng0010, eyptan}@ntu.edu.sg

ABSTRACT

We are often interested to test whether a given cause has a given effect. If we cannot specify the nature of the factors involved, such tests are called model-free studies. There are two major strategies to demonstrate associations between risk factors (ie. patterns) and outcome phenotypes (ie. class labels). The first is that of prospective study designs, and the analysis is based on the concept of "relative risk": What fraction of the exposed (ie. has the pattern) or unexposed (ie. lacks the pattern) individuals have the phenotype (ie. the class label)? The second is that of retrospective designs, and the analysis is based on the concept of "odds ratio": The odds that a case has been exposed to a risk factor is compared to the odds for a case that has not been exposed. The efficient extraction of patterns that have good relative risk and/or odds ratio has not been previously studied in the data mining context. In this paper, we investigate such patterns. We show that this pattern space can be systematically stratified into plateaus of convex spaces based on their support levels. Exploiting convexity, we formulate a number of sound and complete algorithms to extract the most general and the most specific of such patterns at each support level. We compare these algorithms. We further demonstrate that the most efficient among these algorithms is able to mine these sophisticated patterns at a speed comparable to that of mining frequent closed patterns, which are patterns that satisfy considerably simpler conditions.

1. INTRODUCTION

We are often interested to test whether a given cause has a given effect. If we cannot specify the nature of the factors involved, such tests are called model-free studies. There are two major strategies to demonstrate association between risk factors (ie. patterns) and outcome phenotypes (ie. class labels): "cohort study" and "case-control study" [20].

	Disease	No Disease
Exposed	$P_{\mathcal{D},ed}$	$P_{\mathcal{D},e-}$
Not Exposed	$P_{\mathcal{D},-d}$	$P_{\mathcal{D},--}$

Figure 1: Contingency table for a prospective study on cohort $\mathcal{D}$ and risk factor P.

A "cohort study" is a simple way to study whether a specific factor is associated with the risk of a specific disease. Here, cohorts of individuals are identified ones who are exposed, or not exposed, to specific factors. They are then followed prospectively to determine if the exposure affects their risk of disease. The data, in the form of counts, are displayed as a "contingency table", as shown in Figure 1. The analysis is straightforward. A biostatistician simply asks: What is the ratio of the proportions of exposed and unexposed individuals who have developed the disease? This ratio is termed "relative risk". Using the notations in the contingency table in Figure 1, the relative risk of a factor P in a group $\mathcal{D}$ is

$$RR(P,\mathcal{D}) = \frac{\dfrac{P_{\mathcal{D},ed}}{P_{\mathcal{D},ed}+P_{\mathcal{D},e-}}}{\dfrac{P_{\mathcal{D},-d}}{P_{\mathcal{D},-d}+P_{\mathcal{D},--}}}$$

where $P_{\mathcal{D},ed}$ is the number of individuals in $\mathcal{D}$ who have exposure to P and have developed the disease, $P_{\mathcal{D},e-}$ is the number of individuals who have exposure and have not developed the disease, $P_{\mathcal{D},-d}$ is the number of individuals who have no exposure and have developed the disease, and $P_{\mathcal{D},--}$ is the number of individuals who have no exposure and have not developed the disease. Under the null hypothesis, the factor makes no difference to the occurrence of the diease and the relative risk should equal 1. A chi-square test with 1 degree of freedom, $\chi^2(P,\mathcal{D}) = ((P_{\mathcal{D},ed} * P_{\mathcal{D},--} - P_{\mathcal{D},e-} * P_{\mathcal{D},-d})^2 * (P_{\mathcal{D},ed}+P_{\mathcal{D},--}+P_{\mathcal{D},e-}+P_{\mathcal{D},-d}))/((P_{\mathcal{D},ed}+P_{\mathcal{D},-d}) * (P_{\mathcal{D},e-}+P_{\mathcal{D},--}) * (P_{\mathcal{D},ed}+P_{\mathcal{D},e-}) * (P_{\mathcal{D},-d}+P_{\mathcal{D},--}))$, is usually applied to test whether the observed relative risk differs significantly from 1.

We see that the cohort study is essentially a prospective study. That is, we first identify those individuals who are exposed or not exposed to a factor, then test for subsequent de-

[*] Contact author, Fax: +65-6872-5743, Tel: +65-6874-2099.

PODS 2005 June 13-15, 2005, Baltimore, Maryland.
Copyright 2005 ACM 1-59593-062-0/05/06 . . . $5.00.

velopment of the disease. In contrast, a "case-control" study first identifies the individuals by disease status and then tests retrospectively for exposure to a factor. In such a situation, the number of diseased persons (called the "cases") and non-diseased persons (called the "controls") is fixed, and thus the risk that an exposed person will become a disease case cannot be estimated directly. Instead, we can consider the ratio of the odds that a case has been exposed to the odds that a control has been exposed. This ratio is called the "odds ratio". Using the notations in Figure 1, the odds ratio for a factor P in a case-control study $\mathcal{D}$ is

$$OR(P, \mathcal{D}) = \frac{\frac{P_{\mathcal{D},ed}}{P_{\mathcal{D},-d}}}{\frac{P_{\mathcal{D},e-}}{P_{\mathcal{D},--}}}$$

Under the null hypothesis, we expect the odds ratio to equal 1. The chi-squared test with 1 degree of freedom is used to test the significance of deviation of the observed odds ratio from 1. As $OR(P, \mathcal{D})$ lies in the range $[0, \infty]$, it is often transformed to the range $[-1, 1]$ as $(OR(P, \mathcal{D}) - 1)/(OR(P, \mathcal{D}) + 1)$. This transformed value is called Yule's Q [2].

In a traditional data mining and classification situation, we can think of a factor or a combination of factors P as a pattern, an exposure to a factor as an occurrence of the pattern in a transaction, and the disease states as class labels. With such an interpretation in mind, we find that the concept of patterns that have significant relative risk or odds ratio offers a very powerful and statistically sound means for separating meaningful patterns from less meaningful ones.

There are many previous studies [1, 3, 8, 9, 13, 15, 16, 19, 21, 12, 10, 14, 17, 18] on frequent itemsets, their closed patterns, and their generators. There are also a number of studies [6, 11] on emerging patterns and their borders. However, to the best of our knowledge, the mining of odds ratio patterns and relative risk patterns has never been investigated before. The exceptions are the two papers of Tan et al. [17, 18]. They evaluated and compared the "interestingness" of odds ratio and a number of other measures for association mining. They showed that odds ratio, along with Yule's Q and Y measures, satisfied many properties considered desirable for association mining. However, they did not propose efficient sound and complete algorithms for mining these patterns. Furthermore, there is no obvious way to adapt the methods for mining frequent itemsets or emerging patterns to obtain efficient methods for mining odds ratio patterns and relative risk patterns. This is because the concepts of odds ratio and relative risk involve two datasets or a dataset with two class labels, and they also consider both the frequencies of occurrence and non-occurrence of a pattern. In contrast, the concept of frequent itemsets involves only one dataset without class labels and is concerned only with frequency of occurrence. Similarly, while the concept of emerging patterns involves two datasets or a dataset with two class labels, it is concerned only with frequency of occurrence.

The efficient mining of odds ratio patterns and relative risk patterns therefore deserves attention. We investigate in this paper the theoretical properties and the efficient mining of these patterns. We show in Section 2 that the space of odds ratio patterns and relative risk patterns can be systematically stratified into plateaus of convex spaces based their support levels. Exploiting convexity, we formulate in Section 3 a number of algorithms to extract the most general and the most specific of such patterns at each support level. We show that these algorithms are sound and complete; that is, these algorithms can mine all odds ratio patterns and relative risk patterns, and can directly output these patterns in a concise and lossless representation. We compare these algorithms through a few experiments in Section 4, where we also demonstrate that the most efficient among these algorithms is able to mine these sophisticated patterns at a speed comparable to that of mining frequent closed patterns, which are patterns that satisfy considerably simpler conditions.

2. THE SPACE OF ODDS RATIO PATTERNS AND RELATIVE RISK PATTERNS

We first recap some results on equivalence classes, closed patterns, generators, plateaus, and convexity that form the basis for the concise representation and efficient mining of frequent itemsets. Then we prove that while the space of odds ratio patterns and relative risk patterns is not convex, it can be systematically decomposed into a series of plateaus of convex spaces. Then we provide a concise representation of the space of odds ratio patterns and relative risk patterns in terms of the borders—ie. the generators and closed patterns—of these plateaus. This concise representation forms the basis for efficiently mining of odds ratio patterns and relative risk patterns in Section 3.

2.1 Equivalence Classes, Generators, Closed Patterns, and Plateaus

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of distinct literals called "items". An "itemset", or a "pattern", is a set of items. A "transaction" is a non-empty set of items. A "dataset" is a non-empty set of transactions. An itemset P is said to be contained or included in a transaction T if $P \subseteq T$. An itemset P is said to be contained in a dataset $\mathcal{D}$ if there is $T \in \mathcal{D}$ such that $P \subseteq T$.

The "support" of an itemset P in a dataset $\mathcal{D}$, denoted $sup(P, \mathcal{D})$, is the number of transactions in $\mathcal{D}$ that contain P. An itemset P is said to be *frequent* in a dataset $\mathcal{D}$ if $sup(P, \mathcal{D})$ is greater than or equal to a pre-specified threshold ms. Given a dataset $\mathcal{D}$ and a support threshold ms, the collection of all frequent itemsets in $\mathcal{D}$ is called the "space of frequent itemsets", and is denoted by $\mathcal{F}(ms, \mathcal{D})$. We assume in general that $|\mathcal{D}| > ms$.

The space of frequent itemsets can be large. However, this space possesses the nice convexity property, which is very helpful when it comes to concise and lossless representation of the space and its subspaces.

DEFINITION 2.1. *A space $\mathcal{S}$ is said to be convex if, for all $X, Y \in \mathcal{S}$ such that $X \subseteq Y$, it is the case that $Z \in \mathcal{S}$ whenever $X \subseteq Z \subseteq Y$.*

For a convex space $\mathcal{S}$, we define the collection of all "most general" itemsets in $\mathcal{S}$ as a "bound" of $\mathcal{S}$, where an itemset X is most general in $\mathcal{S}$ if there is no proper subset of X in $\mathcal{S}$. Similarly, we define the collection of all "most specific" itemsets as another bound of $\mathcal{S}$, where an itemset X is most specific in $\mathcal{S}$ if there is no proper superset of X in $\mathcal{S}$. We call the former bound the "left bound" of $\mathcal{S}$, denoted $\mathcal{L}$; and the latter bound the "right bound" of $\mathcal{S}$, denoted $\mathcal{R}$. We call the pair of left and right bound the "border" of $\mathcal{S}$, which is denoted by $\langle \mathcal{L}, \mathcal{R} \rangle$. We also define $[\mathcal{L}, \mathcal{R}] = \{Z \mid \exists X \in \mathcal{L}, \exists Y \in \mathcal{R}, X \subseteq Z \subseteq Y\}$. $\langle \mathcal{L}, \mathcal{R} \rangle$ and $[\mathcal{L}, \mathcal{R}]$ are two different notions. Specifically, $[\mathcal{L}, \mathcal{R}] = \mathcal{S}$, but $\langle \mathcal{L}, \mathcal{R} \rangle$ is only a concise representation of the whole space $\mathcal{S}$ in a lossless way. Note that $[\{\emptyset\}, \{\emptyset\}] = \{\emptyset\}$ and $[\emptyset, \emptyset] = \emptyset$.

PROPOSITION 2.2. *$\mathcal{F}(ms, \mathcal{D})$ is convex. Furthermore, its border is of the form $\langle \{\emptyset\}, \mathcal{R} \rangle$ for some $\mathcal{R}$.*

PROOF. *Suppose two itemsets $X, Y \in \mathcal{F}(ms, \mathcal{D})$ such that $X \subseteq Y$. Then $sup(X, \mathcal{D}) \geq sup(Y, \mathcal{D}) \geq ms$. Let Z be any itemset such that $X \subseteq Z \subseteq Y$. Then $sup(Z, \mathcal{D}) \geq sup(Y, \mathcal{D})$. Then $sup(Z, \mathcal{D}) \geq ms$. So, $Z \in \mathcal{F}(ms, \mathcal{D})$. Thus, $\mathcal{F}(ms, \mathcal{D})$ is convex space.*

Since $\emptyset \subset T$ for every $T \in \mathcal{D}$, it is the case that $sup(\emptyset, \mathcal{D}) = |\mathcal{D}| \geq ms$. Therefore, $\emptyset \in \mathcal{F}(ms, \mathcal{D})$. As $\emptyset$ is a proper subset for any non-empty itemsets, it follows that $\emptyset$ is the sole most general element in $\mathcal{F}(ms, \mathcal{D})$. So, the border of $\mathcal{F}(ms, \mathcal{D})$ is of the form $\langle \{\emptyset\}, \mathcal{R} \rangle$, for some $\mathcal{R}$. □

The results stated in Proposition 2.2 are simple and well known. In particular, the Max-Miner algorithm [5] was proposed to discover frequent maximal itemsets ($\mathcal{R}$ in Proposition 2.2) that can cover all frequent itemsets. However, an interesting thing is that convex frequent itemset spaces can be further decomposed systematically into convex subspaces.

DEFINITION 2.3. *An "itemset plateau", $plateau(\pi, \mathcal{D})$, in a dataset $\mathcal{D}$ is defined as the collection of all itemsets that have the support value π in $\mathcal{D}$.*

PROPOSITION 2.4. *$plateau(\pi, \mathcal{D})$ is convex.*

PROOF. *Without loss of generality, let $plateau(\pi, \mathcal{D}) \neq \emptyset$. Suppose itemsets X and Y are in $plateau(\pi, \mathcal{D})$ and $X \subseteq Y$. Then $sup(X, \mathcal{D}) = sup(Y, \mathcal{D}) = \pi$. Let Z be any itemset Z such that $X \subseteq Z \subseteq Y$. Then $sup(X, \mathcal{D}) \geq sup(Z, \mathcal{D}) \geq sup(Y, \mathcal{D})$. As $sup(X, \mathcal{D}) = sup(Y, \mathcal{D}) = \pi$, it follows that $sup(Z, \mathcal{D}) = \pi$. Therefore the itemset Z is in $plateau(\pi, \mathcal{D})$ as desired.* □

As a convex space can be concisely represented by its border, every itemset plateau can be concisely represented by a border. Based on Proposition 2.4, a new concise representation of frequent itemsets is proprosed. Suppose $\mathcal{F}(ms, \mathcal{D})$ is a frequent itemset space in a dataset $\mathcal{D}$, and suppose these frequent itemsets have k distinct support values π_1, π_2, ..., π_k. Then

$$\mathcal{F}(ms, \mathcal{D}) = \bigcup_{i=1}^{k} plateau(\pi_i, \mathcal{D}) = \bigcup_{i=1}^{k} [\mathcal{L}_i, \mathcal{R}_i],$$

where $\langle \mathcal{L}_i, \mathcal{R}_i \rangle$ is the border of $plateau(\pi_i, \mathcal{D})$.

Two itemsets are "equivalent" in the context of a dataset $\mathcal{D}$ if they are included in exactly the same transactions in $\mathcal{D}$. Formally, let us define the "filter", $f(P, \mathcal{D})$, of an itemset P in a dataset $\mathcal{D}$ as $f(P, \mathcal{D}) = \{T \in \mathcal{D} \mid P \subseteq T\}$. Then the "equivalence class" $[P]_{\mathcal{D}}$ of P in a dataset $\mathcal{D}$ is the collection of itemsets defined as $[P]_{\mathcal{D}} = \{Q \mid f(P, \mathcal{D}) = f(Q, \mathcal{D})\}$.

PROPOSITION 2.5. *$[P]_{\mathcal{D}}$ is convex, and the right bound of its border is a singleton set.*

PROOF. *Let $X, Y \in [P]_{\mathcal{D}}$ such that $X \subseteq Y$. Suppose $X \subseteq Z \subseteq Y$. Then $f(Y, D) \subseteq f(Z, D) \subseteq f(X, D)$. Since $X, Y \in [P]_{\mathcal{D}}$, we know $f(Y, D) = f(P, D) = f(X, D)$. So $f(Z, D) = f(P, D)$ and $Z \in [P]_{\mathcal{D}}$. Thus $[P]_{\mathcal{D}}$ is convex as required.*

Now let $\langle \mathcal{L}, \mathcal{R} \rangle$ be the border of $[P]_{\mathcal{D}}$. Suppose $X, Y \in \mathcal{R}$. Then $f(X, \mathcal{D}) = f(P, \mathcal{D}) = f(Y, \mathcal{D})$. This implies $f(X \cup Y, \mathcal{D}) = f(P, \mathcal{D})$. Thus $X \cup Y \in [P]_{\mathcal{D}}$. Since $X, Y \in \mathcal{R}$, by the definition of the right bound of a border, both X and Y are most specific. Then it must be the case that $X = X \cup Y = Y$. Thus $\mathcal{R}$ is a singleton set. □

It is obvious that $sup(X, \mathcal{D}) = sup(Y, \mathcal{D})$ for all $X, Y \in [P]_{\mathcal{D}}$. Therefore, the equivalence class of any itemset P must be a subset of $plateau(\pi, \mathcal{D})$, where $\pi = sup(P, \mathcal{D})$. It now follows from the definition of equivalence classes that $plateau(\pi, \mathcal{D})$ is the union of non-overlapping equivalence classes. In particular, the number of equivalence classes is equal to the number of itemsets in the right bound of the border of $plateau(\pi, \mathcal{D})$.

COROLLARY 2.6. *$plateau(\pi, \mathcal{D}) = [\mathcal{L}_1 \cup \mathcal{L}_2 \cup \cdots \cup \mathcal{L}_r, \{R_1, R_2, \ldots, R_r\}] = \bigcup_{i=1}^{r} [\mathcal{L}_i, \{R_i\}]$, where $\{[R_1]_{\mathcal{D}}, [R_2]_{\mathcal{D}}, ..., [R_r]_{\mathcal{D}}\} = \{[P]_{\mathcal{D}} \mid sup(P, \mathcal{D}) = \pi\}$, and each $\langle \mathcal{L}_i, \{R_i\} \rangle$ is the border of the corresponding $[R_i]_{\mathcal{D}}$.*

Together with equivalence classes, frequent "closed patterns" and frequent "key patterns" (also called "generators") have been widely studied in the data mining field. One main reason is that both of them can be separately used to concisely represent the whole space of frequent itemsets. We discuss next the relationship between plateau boundaries and close and key patterns.

Traditionally, closed patterns are defined based on the notion of equivalence classes or based on changes in support. For example, [15] defines a closed pattern as the maximal itemset of an equivalence class, while [19] defines it equivalently as an itemset that has no proper superset with the same support. As discussed, a $plateau(\pi, \mathcal{D})$ is the collection of all itemsets whose support value in $\mathcal{D}$ is π. The right bound of $plateau(\pi, \mathcal{D})$ are itemsets that are most specific in $plateau(\pi, \mathcal{D})$. That is: any proper superset of any itemset in this bound is not in $plateau(\pi, \mathcal{D})$. So, these proper supersets have a different support level. Hence, the itemsets in the right bound of $plateau(\pi, \mathcal{D})$ are all closed patterns. Conversely, all closed patterns with the same support value π_i are in the right bound of $plateau(\pi_i, \mathcal{D})$. So, the right bounds of our itemset plateaus capture exactly all closed patterns.

The definition of key patterns (generators) is also based on equivalence classes. In particular [4], a pattern P is said to be a key pattern in $\mathcal{D}$ iff no proper subset of P is in $[P]_{\mathcal{D}}$. According to this definition and our definition for the left bound of $plateau(\pi_i, \mathcal{D})$, it can be seen that the itemsets in the left bounds of plateaus are all key patterns. Incidentally, an interesting property for the left bounds of all $plateau(\pi_i, \mathcal{D})$ is that they form a convex space. [4] also observes this property, albeit stated in a different way, and uses it as the basis of an algorithm to find all frequent key patterns.

PROPOSITION 2.7. *Let $\mathcal{F}(ms, \mathcal{D}) = \bigcup_{i=1}^{k} plateau(\pi_i, \mathcal{D}) = \bigcup_{i=1}^{k} [\mathcal{L}_i, \mathcal{R}_i]$, where $\mathcal{L}_i$ and $\mathcal{R}_i$ are bounds of the itemset plateaus. Then $\bigcup_{i=1}^{k} \mathcal{L}_i$ is convex.*

PROOF. *Let $\mathcal{K} = \cup_{i=1}^{k}\mathcal{L}_i$. Suppose $X, Y \in \mathcal{K}$ such that $X \subseteq Y$. Suppose $X \subseteq Z \subseteq Y$. We need to prove that $Z \in \mathcal{K}$. The condition $X \subseteq Z \subseteq Y$ implies four cases: $X \subset Z \subset Y$, $X \subset Z = Y$, $X = Z \subset Y$, and $X = Z = Y$. We only need to prove $Z \in \mathcal{K}$ when $X \subset Z \subset Y$. The other three cases are obvious.*

Let $sup(Z, \mathcal{D}) = \pi_m$. Then $Z \in plateau(\pi_m, \mathcal{D})$. We need to prove $Z \in \mathcal{L}_m$. We prove this by contradiction. Assume $Z \notin \mathcal{L}_m$. Then there is $A \in \mathcal{L}_m$ such that $A \subset Z$ because $plateau(\pi_m, \mathcal{D})$ is a convex space.

Let $Y = Z \cup V$ such that $V \cap Z = \emptyset$. As $A, Z \in plateau(\pi_m, \mathcal{D})$, we have $sup(A, \mathcal{D}) = \pi_m = sup(Z, \mathcal{D})$. As $A \subset Z$, we further have $sup(A \cup V, \mathcal{D}) = sup(Z \cup V, \mathcal{D}) = sup(Y, \mathcal{D})$. So, Y has a proper subset $A \cup V$ with the same support as itself. But Y is an itemset in the left bound of a plateau. So all its proper subsets must have a larger support than Y's. This is a contradiction. So, $Z \in \mathcal{L}_m$. Then $Z \in \mathcal{K}$ as desired. □

The proposition above suggests that if a pattern is in the left bound of a plateau, then all of its subsets are also in the left bound of some other plateaus. In other words, these left bounds (also called key patterns or generators) enjoy the "a priori" property.

PROPOSITION 2.8. *Let P be a frequent generator in $\mathcal{D}$. Then every subset of P is also a frequent generator in $\mathcal{D}$.*

PROOF. *Suppose P is a generator in $\mathcal{D}$. Suppose $Q \subseteq P$. We want to show Q is also a generator. Suppose there is a $R \subseteq Q$ in $[Q]_{\mathcal{D}}$. So there is an S such that $R = Q - S$, and $S \subseteq Q$. Let $T = P - S$. Let D be a transaction having T. We have $R \subseteq T$, as $Q \subseteq P$ and thus $Q - S \subseteq P - S$. Then D has R as well, as $R \subseteq T$. Then D has Q as well, as $R \in [Q]_{\mathcal{D}}$. Then D has S also, as $S \subseteq Q$. Then D has P. This means every transaction having T also has P. We already know that every transaction having P also has T. So, P and T are in the same equivalence class. Since P is a generator, and $T = P - S$, it must be the case that $S = \{\}$. Since $R = Q - S$, we conclude $R = Q$. Hence Q is a key pattern.* □

In the next subsection, we investigate the convexity properties of the space of odds ratio patterns and the space of relative risk patterns.

2.2 Odds Ratio Patterns and Relative Risk Patterns

In order to discuss the space of odds ratio patterns and relative risk patterns, let us consider only those datasets whose transactions are labeled either as "positive" or as "negative". Given a dataset $\mathcal{D}$, we write $\mathcal{D}^{pos}$ for those transactions in $\mathcal{D}$ that are labeled as positive, and $\mathcal{D}^{neg}$ for those labeled as negative. For our discussion on odds ratio and relative risk, we take $\mathcal{D}^{pos}$ to be the disease cases and $\mathcal{D}^{neg}$ to be the non-disease cases.

Since all patterns in $[P]_{\mathcal{D}}$ must occur in exactly the same transactions in $\mathcal{D}$, these patterns must have the same odds ratio, relative risk, and chi-squared value as P. Hence we can extend our notations and write $sup(E, \mathcal{D})$, $OR(E, \mathcal{D})$, $RR(E, \mathcal{D})$, and $\chi^2(E, \mathcal{D})$ respectively for the support, odds ratio, relative risk, and chi-squared value of an equivalent class E in $\mathcal{D}$.

PROPOSITION 2.9. *Let $Q \in [P]_{\mathcal{D}}$. Then $sup(Q, \mathcal{D}) = sup(P, \mathcal{D})$, $OR(Q, \mathcal{D}) = OR(P, \mathcal{D})$, $RR(Q, \mathcal{D}) = RR(P, \mathcal{D})$, and $\chi^2(Q, \mathcal{D}) = \chi^2(P, \mathcal{D})$.*

id	A	B	C	class
1	y	y	y	pos
2	y	y	y	pos
3	y	n	n	pos
4	y	n	n	pos
5	y	y	y	neg
6	y	y	n	neg
7	n	n	n	neg
8	n	n	n	neg

Figure 2: An example dataset with 4 transactions labeled as "pos" and 4 transactions labeled as "neg".

Moreover, if we decompose the space $\mathcal{F}(ms, \mathcal{D})$ into plateaus based on the support levels of patterns in $\mathcal{D}^{pos}$ and the odds ratio or the relative risk of these patterns, each such plateaus is also convex.

THEOREM 2.10. *Let $\mathcal{S}_{n,k}^{OR}(ms, \mathcal{D}) = \{P \in \mathcal{F}(ms,\ \mathcal{D}) \mid P_{\mathcal{D},ed} = n,\ OR(P, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,k}^{OR}(ms, \mathcal{D})$ is convex.*

PROOF. *Let $X \subseteq Y \in \mathcal{S}_{n,k}^{OR}(ms, \mathcal{D})$ and $X \subseteq Z \subseteq Y$. Then $sup(X, \mathcal{D}^{pos}) = sup(Z, \mathcal{D}^{pos}) = sup(Y, \mathcal{D}^{pos}) = n$. Let $h = |\mathcal{D}^{pos}|$. Then $X_{\mathcal{D},ed}/X_{\mathcal{D},-d} = Y_{\mathcal{D},ed}/Y_{\mathcal{D},-d} = Z_{\mathcal{D},ed}/Z_{\mathcal{D},-d} = n/(h-n)$. Let $g = |\mathcal{D}|$. Then $X_{\mathcal{D},--} + X_{\mathcal{D},e-} = Y_{\mathcal{D},--} + Y_{\mathcal{D},e-} = Z_{\mathcal{D},--} + Z_{\mathcal{D},e-} = g - h$. Then $X_{\mathcal{D},--}/X_{\mathcal{D},e-} = (g - h - X_{\mathcal{D},e-})/X_{\mathcal{D},e-} \geq k * (h-n)/n$. Thus $g - h \geq (k * (h-n) * X_{\mathcal{D},e-}/n) + X_{\mathcal{D},e-}$. Since $X \subseteq Z \subseteq Y$, we have $sup(X, \mathcal{D}^{neg}) = X_{\mathcal{D},e-} \geq sup(Z, \mathcal{D}^{neg}) = Z_{\mathcal{D},e-} \geq sup(Y, \mathcal{D}^{neg}) = Y_{\mathcal{D},e-}$. Thus $g - h \geq (k * (h-n) * Z_{\mathcal{D},e-}/n) + Z_{\mathcal{D},e-}$. Then $Z_{\mathcal{D},--}/Z_{\mathcal{D},e-} \geq k*(h-n)/n$. Thus $OR(Z, \mathcal{D}) = (Z_{\mathcal{D},ed}/Z_{\mathcal{D},-d})/(Z_{\mathcal{D},e-}/Z_{\mathcal{D},--}) \geq k$. Then $Z \in \mathcal{S}_{n,k}^{OR}(ms, \mathcal{D})$ as desired.* □

THEOREM 2.11. *Let $\mathcal{S}_{n,k}^{RR}(ms, \mathcal{D}) = \{P \in \mathcal{F}(ms,\ \mathcal{D}) \mid P_{D,ed} = n,\ RR(P, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,k}^{RR}(ms, \mathcal{D})$ is convex.*

PROOF. *Let $X \subseteq Y \in \mathcal{S}_{\mathcal{D},n,k}^{RR}$ and $X \subseteq Z \subseteq Y$. Then $sup(X, \mathcal{D}^{pos}) = sup(Z, \mathcal{D}^{pos}) = sup(Y, \mathcal{D}^{pos}) = n$. Let $h = |\mathcal{D}^{pos}|$. Then $RR(X, \mathcal{D}) = (n/(h-n))*((h-n)+X_{\mathcal{D},--})/(n+X_{\mathcal{D},e-}) \geq k$. So, $X_{\mathcal{D},--} - ((k/n) * X_{\mathcal{D},e-}) \geq k$. Since $X \subseteq Z$, we have $X_{\mathcal{D},e-} \geq Z_{\mathcal{D},e-}$ and $X_{\mathcal{D},--} \leq Z_{\mathcal{D},--}$. Then $Z_{\mathcal{D},--} - ((k/n) * Z_{\mathcal{D},e-}) \geq X_{\mathcal{D},--} - ((k/n) * X_{\mathcal{D},e-}) \geq k$. So $RR(Z, \mathcal{D}) \geq k$. Thus $Z \in \mathcal{S}_{n,k}^{RR}(ms, \mathcal{D})$ as desired.* □

Since the intersection of two convex spaces is convex, the space of patterns that simultaneously exhibit good odds ratio and relative risk is also convex.

COROLLARY 2.12. *Let $\mathcal{S}_{n,h,k}^{OR,RR}(ms, \mathcal{D}) = \{P \in \mathcal{F}(ms,\ \mathcal{D}) \mid P_{D,ed} = n,\ OR(P, \mathcal{D}) \geq h,\ RR(P, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,h,k}^{OR,RR}(ms, \mathcal{D})$ is convex.*

However, it is not possible to decompose the space of odds ratio patterns into convex subspaces in terms of odds ratio or relative risk alone.

PROPOSITION 2.13. *Let $\mathcal{S}_{k}^{OR}(ms, \mathcal{D}) = \{P \in \mathcal{F}(ms,\ \mathcal{D}) \mid OR(P, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{k}^{OR}(ms, \mathcal{D})$ is not convex.*

Proof. *Consider the dataset $\mathcal{D}$ in Figure 2. Here, $OR(\{A\}, \mathcal{D}) = \infty$, $OR(\{A, B, C\}, \mathcal{D}) = 3$, and $OR(\{A, B\}, \mathcal{D}) = 1$. Then $\{A\}, \{A, B, C\} \in \mathcal{S}_2^{OR}(2, \mathcal{D})$ and $\{A, B\} \notin \mathcal{S}_2^{OR}(2, \mathcal{D})$, giving a counter example.* □

Proposition 2.14. *Let $\mathcal{S}_k^{RR}(ms, \mathcal{D}) = \{P \in \mathcal{F}(ms,\ \mathcal{D}) \mid RR(P, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_k^{RR}(ms, \mathcal{D})$ is not convex.*

Proof. *Consider the dataset $\mathcal{D}$ in Figure 2. Here, $RR(\{A\}, \mathcal{D}) = \infty$, $RR(\{A, B, C\}, \mathcal{D}) = 5/3$, and $RR(\{A, B\},\ \mathcal{D}) = 1$. Then $\{A\}, \{A, B, C\} \in \mathcal{S}_{1.5}^{RR}(2, \mathcal{D})$ and $\{A, B\} \notin \mathcal{S}_{1.5}^{OR}(2, \mathcal{D})$, giving a counter example.* □

Since $\mathcal{S}_{n,k}^{OR}(ms, \mathcal{D})$, $\mathcal{S}_{n,k}^{RR}(ms, \mathcal{D})$, and $\mathcal{S}_{n,h,k}^{OR,RR}(ms, \mathcal{D})$ are convex spaces, they can be represented by borders. There are two approaches to the extraction of these borders. The first approach is to first identify all the equivalence classes and their borders in $\mathcal{D}$, and filter them by their support in $\mathcal{D}^{pos}$ and by their odds ratio and/or relative risk. This approach is summarised by the following two propositions:

Proposition 2.15. *Let $\mathcal{E}_{\mathcal{D}}$ be all the equivalence classes in $\mathcal{D}$. Let $\mathcal{B}_{n,k}^{OR}(ms, \mathcal{D}) = \{\langle \min E,\ \max E\rangle \mid E \in \mathcal{E}_{\mathcal{D}}, sup(E, \mathcal{D}) \geq ms, sup(E,\ \mathcal{D}^{pos}) = n,\ OR(E, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,k}^{OR}(ms,\ \mathcal{D}) = \bigcup\{[\mathcal{L}, \mathcal{R}] \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,k}^{OR}(ms, \mathcal{D})\} = [\bigcup\{\mathcal{L} \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,k}^{OR}(ms, \mathcal{D})\}, \bigcup\{\mathcal{R}|\ \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,k}^{OR}(ms, \mathcal{D})\}]$.*

Proposition 2.16. *Let $\mathcal{E}_{\mathcal{D}}$ be all the equivalence classes in $\mathcal{D}$. Let $\mathcal{B}_{n,k}^{RR}(ms, \mathcal{D}) = \{\langle \min E,\ \max E\rangle \mid E \in \mathcal{E}_{\mathcal{D}}, sup(E, \mathcal{D}) \geq ms, sup(E,\ \mathcal{D}^{pos}) = n,\ RR(E, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,k}^{RR}(ms,\ \mathcal{D}) = \bigcup\{[\mathcal{L}, \mathcal{R}] \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,k}^{RR}(ms, \mathcal{D})\} = [\bigcup\{\mathcal{L} \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,k}^{RR}(ms, \mathcal{D})\}, \bigcup\{\mathcal{R} \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,k}^{RR}(ms, \mathcal{D})\}]$.*

Corollary 2.17. *Let $\mathcal{E}_{\mathcal{D}}$ be all the equivalence classes in $\mathcal{D}$. Let $\mathcal{B}_{n,h,k}^{OR,RR}(ms, \mathcal{D}) = \{\langle \min E,\ \max E\rangle \mid E \in \mathcal{E}_{\mathcal{D}}, sup(E, \mathcal{D}) \geq ms, sup(E,\ \mathcal{D}^{pos}) = n,\ OR(E, \mathcal{D}) \geq h,\ RR(E, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,h,k}^{OR,RR}(ms,\ \mathcal{D}) = \bigcup\{[\mathcal{L}, \mathcal{R}] \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,h,k}^{OR,RR}(ms, \mathcal{D})\} = [\bigcup\{\mathcal{L} \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,h,k}^{OR,RR}(ms, \mathcal{D})\}, \bigcup\{\mathcal{R} \mid \langle \mathcal{L}, \mathcal{R}\rangle \in \mathcal{B}_{n,h,k}^{OR,RR}(ms, \mathcal{D})\}]$.*

The second approach is to first obtain a plateau decomposition of the frequent itemsets in $\mathcal{D}^{pos}$, then extract the border of the plateau corresponding to the support level of n in $\mathcal{D}^{pos}$. According to Corollary 2.6, each element R_j in the right bound of this plateau is the closed pattern of an equivalence class. For each R_j, we check also $\mathcal{D}^{neg}$ to see if this R_j have good odds ratio or good relative risk. If it does not have good odds ratio or good relative risk, then its entire equivalence class can be discarded. If it has good odds ratio or relative risk, then we need to consider the corresponding left bound L_j. By Theorems 2.10 and 2.11, we need to move L_j right to a more specialised L'_j so that the requirements on odds ratio and relative risk are satisfied.

Proposition 2.18. *Let $\bigcup_i plateau(\pi_i, \mathcal{D})$ be the plateau decomposition of $\mathcal{F}(ms, \mathcal{D}^{pos})$. Let $\langle \mathcal{L}_n, \mathcal{R}_n\rangle$ be the border of $plateau(n, \mathcal{D})$. Let $\mathcal{R}'_n = \{R_j \in \mathcal{R}_n \mid OR(R_j, \mathcal{D}) \geq k\}$. Let $\mathcal{L}'_n = \bigcup_{R_j \in \mathcal{R}'_n} \min\{B \subseteq R_j \mid L_j \in \mathcal{L}_n,\ L_j \subseteq B,\ OR(B, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,k}^{OR}(ms, \mathcal{D}) = [\mathcal{L}'_n, \mathcal{R}'_n]$.*

Proposition 2.19. *Let $\bigcup_i plateau(\pi_i, \mathcal{D})$ be the plateau decomposition of $\mathcal{F}(ms, \mathcal{D}^{pos})$. Let $\langle \mathcal{L}_n, \mathcal{R}_n\rangle$ be the border of $plateau(n, \mathcal{D})$. Let $\mathcal{R}'_n = \{R_j \in \mathcal{R}_n \mid RR(R_j, \mathcal{D}) \geq k\}$. Let $\mathcal{L}'_n = \bigcup_{R_j \in \mathcal{R}'_n} \min\{B \subseteq R_j \mid L_j \in \mathcal{L}_n,\ L_j \subseteq B,\ RR(B, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,k}^{RR}(ms, \mathcal{D}) = [\mathcal{L}'_n, \mathcal{R}'_n]$.*

```
Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, threshold for support $ms$,
    and threshold for odds ratio $k$.
Output: A concise representation of equivalence classes of
    patterns having support at least $ms$ and odds ratio at
    least $k$. The concise representation comprises, for each
    equivalence class, its borders (ie. its generators and
    closed patterns), its support in $\mathcal{D}^{pos}$, its support in $\mathcal{D}^{neg}$,
    and its odds ratio.
Method:
 1: $\mathcal{E}$ := the collection of all equivalence classes of $\mathcal{F}(ms, \mathcal{D})$,
    concisely represented by their borders, and annotated
    with their support levels.
 2: for each $\langle \mathcal{L}, \{R\}\rangle$ in $\mathcal{E}$ do
 3:     $x := OR(R, \mathcal{D})$;
 4:     $y := sup(R, \mathcal{D}^{pos})$; $z := sup(R, \mathcal{D}^{neg})$.
 5:     if $x \geq k$ then
 6:         output $\langle \mathcal{L}, \{R\}\rangle$, $x$, $y$, and $z$.
 7:     end if
 8: end for
```

Figure 3: Algorithm for mining odds ratio patterns.

Corollary 2.20. *Let $\bigcup_i plateau(\pi_i, \mathcal{D})$ be the plateau decomposition of $\mathcal{F}(ms, \mathcal{D}^{pos})$. Let $\langle \mathcal{L}_n, \mathcal{R}_n\rangle$ be the border of $plateau(n, \mathcal{D})$. Let $\mathcal{R}'_n = \{R_j \in \mathcal{R}_n \mid OR(R_j, \mathcal{D}) \geq h,\ RR(R_j, \mathcal{D}) \geq k\}$. Let $\mathcal{L}'_n = \bigcup_{R_j \in \mathcal{R}'_n} \min\{B \subseteq R_j \mid L_j \in \mathcal{L}_n,\ L_j \subseteq B,\ OR(B, \mathcal{D}) \geq h,\ RR(B, \mathcal{D}) \geq k\}$. Then $\mathcal{S}_{n,h,k}^{RR}(ms, \mathcal{D}) = [\mathcal{L}'_n, \mathcal{R}'_n]$.*

In the interest of space, for remainder of this paper, we focus on evaluating a couple of implementations for identifying good odds ratio patterns based on the strategy in Proposition 2.15.

3. PRACTICAL ALGORITHMS

Proposition 2.15 suggests the algorithm in Figure 3 for mining all patterns that have support of at least ms and odds ratio of at least k. It is also clear that all the steps of this algorithm are simple, with the exception of Step 1, for generating the concise representation—comprising of the generators and the unique closed pattern of each equivalence class—of all equivalence classes in $\mathcal{F}(ms, \mathcal{D})$. Note that by replacing Step 3 of this algorithm with $x := RR(R, \mathcal{D})$, we get an algorithm for mining relative risk patterns.

The mining of close patterns has previously been studied intensively [3, 8, 9, 13, 15, 16, 19, 21]. The mining of generators has also previously been investigated [12, 14, 15], albeit to a less intensive extent. There is however not much reported work on algorithms that produce both generators and closed patterns, with the exception of [14].

We therefore have two options for implementing Step 1 of the odds ratio mining algorithm of Figure 3. This first option is to apply on $\mathcal{D}$ a fast closed-pattern mining algorithm and a fast generator mining algorithm to produce separately a list of closed patterns C_1, ..., C_m and a list of generators G_1, ..., G_n, as well as their support levels. By Proposition 2.5, each C_i is the closed pattern of a distinct equivalence class. The corresponding generators of the equivalence class of C_i are those G_j such that $sup(G_j, \mathcal{D}) = sup(C_i, \mathcal{D})$ and $G_j \subseteq C_i$. In this case, we use FPclose* [9] to mine frequent closed patterns, as it is currently the fastest known program for this problem. For the mining of generators, none of the existing methods is sufficiently efficient. So we develop our own method, *Gr-growth*, that leverages the infrastructure of FPclose*. This option is pursued in Subsection 3.1.

The second option is to mine generators and closed patterns at the same time. However, the only reported method that can be easily modified to provide generators and closed patterns simultaneously, A-Close [14], is not sufficiently efficient. So we develop our own technique, *GC-growth*. This option is pursued in Subsection 3.2. This second option is likely to have an advantage over the first option since it avoids the sorting and matching step that is needed in the first option to properly pair up the generators with the corresponding closed patterns.

3.1 Mining Generators and Mining Closed Patterns Separately

In this subsection, we present the *Gr-growth* algorithm for mining generators that is used for the first option described earlier for mining odds ratio patterns. We also prove its correctness.

Let the set $I = \{i_1, ..., i_m\}$ of items be ordered according to an arbitrary ordering $<_0$ so that $i_1 <_0 i_2 <_0 \cdots <_0 i_m$. For itemsets X, $Y \subseteq I$, we write $X <_0 Y$ iff X is lexicographically "before" Y according to the order $<_0$. We say an itemset X is a "prefix" of an itemset Y iff $X \subseteq Y$ and $X <_0 Y$. We say an itemset X is a "postfix" of an itemset Y iff $X \subseteq Y$ and $Y - X <_0 Y$. We write $last(X)$ for the item $\alpha \in X$ such that $\{\alpha\}$ is postfix of the itemset X; more intuitively, if the items in X are $\alpha_1 <_0 \alpha_2 <_0 \cdots <_0 \alpha_m$, then $last(X) = \alpha_m$.

A set-enumeration tree is a conceptual organization on the subsets of I so that $\{\}$ is its root node; for each node X such that Y_1, ..., Y_k are all its children from left to right, then $Y_1 <_0 \cdots <_0 Y_k$; for each node X in the set-enumeration tree such that X_1, ..., X_k are siblings to its right, we make $X \cup X_1$, ..., $X \cup X_k$ the children of X; $|X \cup X_i| = |X| + 1 = |X_i| + 1$; and $|X| = |X_i| = |X \cap X_i| + 1$.

We also induce an enumeration ordering on the nodes of this set-enumeration tree so that given two nodes X and Y, we say $X <_1 Y$ iff X would be visited before Y when we visit the set-enumeration tree in a right-to-left top-down manner. We say X "precedes" Y if $X <_1 Y$. Since this visit order is a bit unusual, we illustrate it in Figure 4. Here, the number besides the node indicates the time at which the node is visited.

Proposition 3.1. *Let X and Y be nodes on the set-enumeration tree so that $X \subseteq Y$. Then $X <_1 Y$.*

We use the set-enumeration tree as a conceptual structure so that given any pattern X, we can efficiently test the presence of X, the number of occurrences of X, and the transactions in which X appears fast. In particular, for each node X of the set-enumeration tree visited for a dataset $\mathcal{D}$, we associate the following maps, where d_T is the transaction id of the transaction T in $\mathcal{D}$:

- $P[X] = |\{d_T \mid T \in \mathcal{D}, X \text{ is a prefix of } T\}|$;
- $P^{pos}[X] = |\{d_T \mid T \in \mathcal{D}^{pos}, X \text{ is a prefix of } T\}|$;
- $P^{neg}[X] = |\{d_T \mid T \in \mathcal{D}^{neg}, X \text{ is a prefix of } T\}|$;
- $S[X] = sup(X, \mathcal{D})$;

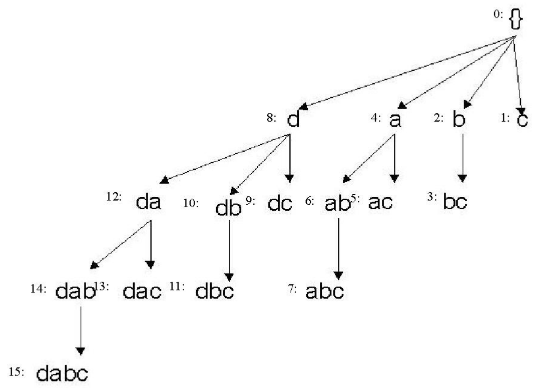

Figure 4: A set-enumeration tree depicting the subsets of $\{a, b, c, d\}$, with $d <_0 a <_0 b <_0 c$.

- $S^{pos}[X] = sup(X, \mathcal{D}^{pos})$;
- $S^{neg}[X] = sup(X, \mathcal{D}^{neg})$;
- $T[X]$ = true iff X is an entire transaction;
- $G[X]$ = true iff X is a generator in $\mathcal{D}$; and
- $H[\alpha] = \{X | P[X] \text{ is defined}, \{\alpha\} \text{ is suffix of } X\}$.

Provided these maps are constructed correctly, it is clear that $\{X \mid G[X] = \text{true}\}$ gives all the generators in $\mathcal{D}$. We provide in Figure 5 the pseudo codes of an efficient algorithm, *Gr-growth*, for mining generators.

Theorem 3.2. *Given a dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$ and a support threshold ms, Gr-growth is sound and complete for producing the generators of $\mathcal{F}(ms, \mathcal{D})$ and their support levels.*

Proof. *Without loss of generality, we assume Step 1 is correct. Then, for any node X_i, $P[X_i]$ (resp. $P^{pos}[X_i]$, $P^{neg}[X_i]$) gives the number of occurrences of X_i in $\mathcal{D}$ (resp. $\mathcal{D}^{pos}$, $\mathcal{D}^{neg}$) as a prefix. Then, for any node X_i, the set $\{X \in H[last(X_i)] \mid X_i \subseteq X\}$ comprises precisely those prefixes that contain X_i. So $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$ gives the number of occurrences of X_i in $\mathcal{D}$. Thus, $S[X_i] = sup(X_i, \mathcal{D})$. Similarly, we conclude $S^{pos}[X_i] = sup(X_i, \mathcal{D}^{pos})$ and $S^{neg}[X_i] = sup(X_i, \mathcal{D}^{neg})$. Thus Steps 4–5 are correct. By Proposition 2.8, X_i is a generator iff $sup(X_i) \geq ms$, $sup(X_i) < sup(X_i - \{\alpha\}, \mathcal{D})$ for all $\alpha \in X_i$, and $X_i - \{\alpha\}$ is a generator for all $\alpha \in X_i$. Thus Steps 7–10 are correct, provided that $S[X_i - \{\alpha\}]$ and $G[X_i - \{\alpha\}]$ are already computed at this point. By Proposition 3.1, and the order given in Step 2 of Gr-growth, this is indeed the case. Thus Gr-growth is sound. Since, Step 2 of Gr-growth enumerates all possible itemsets, Gr-growth is complete.* □

There are a number of practical matters involved in getting an efficient implementation of *Gr-growth*. First, we run *Gr-growth* to mine generators after we run FPclose* to mine closed patterns. FPclose* [9] constructs a special prefix tree structure that is very much like the classical trie structure [7], as well as a head table that maps every item to

```
Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold
    $ms$.
Output: The generators of $\mathcal{F}(ms, \mathcal{D})$ and their support lev-
    els.
Method:
 1: Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, $T[\cdot]$, and $H[\cdot]$ by making one
    pass through $\mathcal{D}$; set $G[\{\}] :=$ true.
 2: Let $X_1$, ..., $X_n$ be the nodes in the set-enumeration
    tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
 3: for $i = 1, .., n$ do
 4:     $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
 5:     $S^{pos}[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
 6:     $S^{neg}[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
 7:     if $S[X_i] \geq ms$ and $\forall\alpha \in X_i . S[X_i] < S[X_i - \{\alpha\}]$ and
        $\forall\alpha \in X_i . G[X_i - \{\alpha\}] =$ true then
 8:         $G[X_i] :=$ true;
 9:         output $X_i$, $S^{pos}[X_i]$, and $S^{neg}[X_i]$;
10:     else
11:         $G[X_i] :=$ false;
12:     end if
13: end for
```

Figure 5: Pseudo codes for *Gr-growth*.

all its occurrences in the prefix tree. This special prefix tree is isomorphic[1] to the set-enumeration tree if we map each node X in the set-enumeration tree to a corresponding node whose path is X in the prefix tree. We modify FPclose* so that $P[X]$, $P^{pos}[X]$, $P^{neg}[X]$, and $T[X]$ are efficiently computed and stored at the node whose path is X in FPclose*'s prefix tree. In this case, $T[X]$ is just a flag in the last node of the path X in FPclose*'s prefix tree to mark if the path X is an entire transaction. The map $H[\cdot]$ then corresponds to the head table constructed by FPclose*.

Second, although we use a for-loop in Step 3 of the pseudo codes of *Gr-growth*, we traverse in reality the prefix tree generated by FPclose* in an order corresponding to a depth-first right-to-left traversal of the set-enumeration tree. By Proposition 2.8, if X_i is not a generator, then all its supersets cannot be generators. By definition of the set-enumeration tree, all the supersets of X_i are enumerated in the subtrees of X_i and subtrees of X_i's siblings to the left. Therefore, by the isomorphism between our set-enumeration tree and the prefix tree constructed by FPclose*, we can easily skip traversing all the subtrees and eliminate the computations for X_i's supersets.

Third, to avoid walking up and down the prefix tree when looking for $S[X]$, $S^{pos}[X]$, $S^{neg}[X]$, and $G[X]$, we store pointers to the node corresponding to X in a hashtable indexed by a hash function $\kappa(P) = \sum_{i \in P} 2^{\hat{i}-1} mod\ q$, where q is the hashtable size, and $\hat{i} = j$ if i is the jth item in $I = \{i_1, ..., i_m\}$ according to the ordering $<_0$. Incidentally, $X <_1 Y$ iff $\sum_{i \in X} 2^{\hat{i}-1} < \sum_{i \in Y} 2^{\hat{i}-1}$.

3.2 Mining Generators and Mining Closed Patterns Together

In this subsection, we present the *GC-growth* algorithm for mining generators and closed patterns simultaneously that is used for the second option described earlier for mining odds ratio patterns. We also prove its correctness.

To implement *GC-growth*, we observe that

PROPOSITION 3.3. *Let a dataset $\mathcal{D}$ be given. Let $P[X] = |\{d_T \mid T \in \mathcal{D}, X$ is prefix of $T\}|$. Let $H[\alpha] = \{X \mid P[X]$ is defined, $\{\alpha\}$ is suffix of $X\}$. Let X be a generator in $\mathcal{D}$. Then the closed pattern of $[X]_{\mathcal{D}}$ is $\bigcap\{X'' \mid X' \in H[last(X)]$, $X \subseteq X'$, X' is prefix of X'', $T[X''] = true\}$.*

PROOF. *Let X be a generator. Let C be the unique closed pattern of the equivalence class of X. Then C is in every transaction T that contains X. Let $X' \in H[last(X)]$ such that $X \subseteq X'$. Then C is in every transaction T that contains X'. By construction, $S_{X'} = \{X'' \mid X'$ is prefix of X'', $T[X''] = true\}$ are precisely those transactions having X' as a prefix. In other words, $S = \bigcup_{X' \in H[last(X)], X \subseteq X'} S_{X'} = f(X, \mathcal{D})$. Since C is a closed pattern of $[X]_{\mathcal{D}}$, it is the largest itemset that is common to all transactions in S. Then $C = \bigcap S$.* □

Therefore, to simultaneously mine generators and closed patterns, it suffices to make a modification to the *Gr-growth* algorithm presented earlier to incorporate Proposition 3.3. The pseudo codes of the resulting algorithm, *GC-growth* is presented in Figure 6. We use $S^{pos}[\cdot]$ and $S^{neg}[\cdot]$ to store the support of closed patterns—rather than generators—in $\mathcal{D}^{pos}$ and $\mathcal{D}^{neg}$. We also use an additional map $R[\cdot]$, so that $R[C]$ is the set of generators corresponding to closed pattern C. The correctness of this algorithm follows easily from Proposition 3.3 and the proof of Theorem 3.2.

THEOREM 3.4. *Given a dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$ and a support threshold ms, GC-growth is sound and complete for producing the generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$ and their support levels simultaneously.*

There are a number of practical matters involved in getting an efficient implementation of *GC-growth*. First, as in *Gr-growth*, we use a special prefix tree and head table to keep $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, $T[\cdot]$, and $H[\cdot]$. Note that in the case of *Gr-growth*, FPclose* is run first to produce closed patterns, and the prefix tree and head table are produced as a byproduct by FPclose*. In the case of *GC-growth*, we only run the part of FPclose* that builds the prefix tree and head table, but we do not run the rest of FPclose*.

Second, as in *Gr-growth*, although we use a for-loop in Step 3 of the pseudo codes of *GC-growth*, we traverse in reality the prefix tree. As before, by Proposition 2.8, we skip the traversal and computations involving supersets of those X_i that are not generators.

Third, as in *Gr-growth*, to avoid walking up and down the prefix tree when looking for $S[X]$ and $G[X]$, we use a hashtable. Similarly, $R[C]$ is implemented as a hashtable.

Fourth, we optimize the computation of $S = \bigcap\{X'' \mid X' \in H[last(X_i)]$, $X_i \subseteq X'$, X' is prefix of X'', $T[X''] = true\}$. Note that in Step 4 of *GC-growth*, we have already identified $S' = \{X' \in H[last(X_i)] \mid X_i \subseteq X'\}$. We can thus re-use this in the computation of S, and avoid traversing those branches of the prefix tree that do not contain X_i. Furthermore, the

[1] This isomorphism is partial in the sense nodes that correspond to those itemsets that do not occur in the dataset are not mapped. However, it is not necessary to store any information for these nodes, since the information is either not needed or can be inferred.

```
Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold $ms$.
Output: The generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$, as well as and their support levels.
Method:
Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, $T[\cdot]$, and $H[\cdot]$ by making one pass through $\mathcal{D}$; set $G[\{\}] :=$ true; initialize $R[\cdot]$ to empty.
Let $X_1, ..., X_n$ be the nodes in the set-enumeration tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
for $i = 1, .., n$ do
    $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
    if $S[X_i] \geq ms$ and $\forall \alpha \in X_i.S[X_i] < S[X_i - \{\alpha\}]$ and $\forall \alpha \in X_i.G[X_i - \{\alpha\}] =$ true then
        $G[X_i] :=$ true;
        $C := \bigcap\{X'' \mid X' \in H[last(X_i)], X_i \subseteq X', X'$ is prefix of $X''$, $T[X''] = true\}$;
        $R[C] := R[C] \cup \{X_i\}$;
        $S^{pos}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
        $S^{neg}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
    else
        $G[X_i] :=$ false;
    end if
end for
for each $C$ such that $R[C] \neq \{\}$ do
    output $R[C]$, $C$, $S^{pos}[C]$, and $S^{neg}[C]$;
end for
```

Figure 6: Pseudo codes for *GC-growth*.

$\bigcap$ computation can be avoided by walking down the subtree of the node corresponding to each X', and keeping those items that have a total $P[\cdot]$ count that is equal to $P[X']$.

4. PERFORMANCE STUDIES

We next conduct a few experiments for mining odds ratio patterns on a number of benchmark datasets—mushroom, connect-4, and chess—stored at the FIMI Repository, `http://fimi.cs.helsinki.fi`. We use a PC with P4 CPU, 2.4GHz, and 512MB RAM.

The mining results are shown in Figure 7. It contains the timing information of the two options as discussed in Section 3 for mining closed patterns and generators. This figure also contains the number of qualified closed patterns and generators under different support thresholds and different odds ratio thresholds from the three datasets.

It is clear from Figure 7 that the second option (mining generators and closed patterns simultaneously) is consistently faster than the first option (mining generators and closed patterns separately). The speed-up becomes more significant when the threshold of the odds ratio decreases from 10 to 2.5. There are two main reasons. The first reason is that significant extra computations are needed for sorting and grouping the separately mined generators with the right closed patterns. The second reason is that the mining the generators via *Gr-growth* in a separate process from mining the closed patterns via FPclose* causes the prefix tree to be traversed and processed twice.

We also found that our speed (by the second option) of mining odd ratio patterns is comparable to that of mining closed patterns by state-of-the-art algorithms FPClose* [9] and CLOSET+ [19]. Detailed timing information is shown in Figure 8.

5. CONCLUDING REMARKS

In this paper, we introduce two very important types of patterns that are commonly used in clinical studies and other real-world applications—viz. odds ratio patterns and relative risk patterns [20]. To the best of our knowledge, these patterns have not be previously investigated by data mining researchers.

We study the theoretical properties of the space of odds ratio patterns and relative risk patterns. We show that these two pattern spaces are not convex. However, we also show that there is a systematic way to decompose these two pattern spaces into a series of plateaus, based on support levels, such that each plateau is convex. Therefore, each plateau can be concisely represented by a border comprising the generators and the unique closed pattern of the plateau.

Then we present two algorithms to efficiently mine odds ratio patterns and relative risk patterns. The first one mines the generators and closed patterns of equivalence classes separately, and then combines them to discover odds ratio patterns and relative risk patterns. The second one mines the generators and closed patterns of equivalence classes at the same time, and then uses them to discover odds ratio patterns and relative risk patterns. At the core of these algorithms are two novel methods—*Gr-growth* for mining generators, and *GC-growth* for simultaneously mining generators and closed patterns.

Finally, we perform a number of experiments, and show that both algorithms are efficient. However, the first algorithm is considerably less efficient than the second one, as it is dominated by the cost of sorting and matching the generators to the corresponding closed patterns. We also show that the second algorithm is able to produce odds ratio patterns and relative risk patterns at a cost comparable to the mining of closed patterns, which are a considerably simpler type of patterns.

Correction Notes

This is the corrected version of our paper in PODS 2006. In the version that appears in PODS 2006, Proposition 3.3 and the pseudo codes for *Gr-growth* and *GC-growth* are given incorrectly.

6. REFERENCES

[1] R. Agrawal, et al. Mining association rules between sets of items in large databases. In *Proceedings of 12th ACM-SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[2] A. Agresti. *An Introduction to Categorical Data Analysis*. Wiley & Sons, New York, 1996.

[3] Y. Bastide, et al. Mining minimal non-redundant association rules using frequent closed itemsets. In *Computational Logic*, pages 972–986, 2000.

[4] Y. Bastide, et al. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2:66–75, 2000.

[5] R. J. Bayardo. Efficiently mining long patterns from databases. In *Proceedings of 17th ACM-SIGMOD International Conference on Management of Data*, pages 85–93, 1998.

[6] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 15–18, 1999.

[7] E. Fredkin. Trie memory. *Communications of ACM*, 3:490–500, 1960.

[8] B. Goethals and M. J. Zaki. FIMI03: Workshop on frequent itemset mining implementations. In *Proceedings of ICDM2003 Workshop on Frequent Itemset Mining implementations*, pages 1–13, 2003.

[9] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *Proceedings of ICDM2003 Workshop on Frequent Itemset Mining Implementations*, 2003.

[10] J. Han, et al. Mining frequent patterns without candidates generation. In *Proceedings of 19th ACM-SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[11] J. Li, et al. The space of jumping emerging patterns and its incremental maintenance algorithms. In *Proceedings of 17th International Conference on Machine Learning*, pages 551–558, 2000.

[12] V. P. Luong. The closed keys base of frequent itemsets. In *Proceedings of 4th International Conference on Data Warehousing and Knowledge Discovery*, pages 181–190, 2002.

[13] F. Pan, et al. CARPENTER: Finding closed patterns in long biological datasets. In *Proceedings of 9th ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 637–642, 2003.

[14] N. Pasquier, et al. Discovering frequent closed itemsets for association rules. In *Proceedings of 7th International Conference on Database Theory*, pages 398–416, 1999.

[15] N. Pasquier, et al. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24:25–46, 1999.

[16] J. Pei, et al. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *Proceedings of ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[17] P.-N. Tan, et al. Selecting the right interestingness measure for association patterns, In *Proceedings of 8th ACM-SIGKDD International Conference on Knowledge Dicovery and Data Mining*, pages 32–41, 2002.

[18] P.-N. Tan, et al. Selecting the right objective measure for association analysis, *Information Systems*, 29:293–313, 2004.

[19] J. Wang, et al. CLOSET+: Search for the best strategies for mining frequent closed itemsets. In *Proceedings of 9th ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 236–245, 2003.

[20] K. M. Weiss. *Genetic Variation and Human Disease: Principles and Evolutionary Approaches.* Cambridge University Press, 1993.

[21] M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In *Proceedings of 2nd SIAM International Conference on Data Mining*, pages 457–473, 2002.

mushroom				
OR	Support	0.1%	1.0%	2.0%
2.5	# closed patterns	48247	10428	5832
	# generators	95744	24322	12181
	first approach (secs)	53.410	11.922	4.891
	second approach (secs)	3.359	1.281	0.921
5.0	# closed patterns	44343	8186	4477
	# generators	89298	21087	10376
	first approach (secs)	52.745	10.212	4.860
	second approach (secs)	3.296	1.25	0.906
10	# closed patterns	41799	6662	3350
	# generators	85492	18959	8845
	first approach (secs)	52.477	9.988	4.815
	second approach (secs)	3.265	1.234	0.875

chess				
OR	Support	30.0%	40.0%	50.0%
2.5	# closed patterns	795454	180764	40598
	# generators	818134	181083	40609
	first approach (secs)	6551	408.228	51.064
	second approach (secs)	766.593	13.796	3.468
5.0	# closed patterns	91389	18995	7753
	# generators	91525	18995	7753
	first approach (secs)	3659	140.373	15.987
	second approach (secs)	515.109	12.015	3.171
10	# closed patterns	28233	1591	21
	# generators	28233	1591	21
	first approach (secs)	647.549	34.326	6.052
	second approach (secs)	543.015	11.937	3.437

connect-4				
OR	Support	20.0%	30.0%	40.0%
2.5	# closed patterns	1	0	0
	# generators	1	0	0
	first approach (secs)	32.933	13.125	8.750
	second approach (secs)	10.312	3.453	2.328
5.0	# closed patterns	0	0	0
	# generators	0	0	0
	first approach (secs)	33.011	12.484	8.672
	second approach (secs)	10.203	4.406	2.312
10	# closed patterns	0	0	0
	# generators	0	0	0
	first approach (secs)	32.956	12.625	8.848
	second approach (secs)	10.265	3.593	2.25

Figure 7: The number of closed patterns and generators from the three datasets that satisfy different levels of support thresholds and odds ratio thresholds, and the timing information by the two mining approaches.

mushroom			
Support	0.1%	1.0%	2.0%
FPClose* (secs)	2.468	0.703	0.421
CLOSET+ (secs)	5.547	1.156	0.657
Our method (secs)	3.359	1.281	0.921

chess			
Support	30.0%	40.0%	50.0%
FPClose* (secs)	96.5	18.546	4.234
CLOSET+ (secs)	1244.250	76.610	8.328
Our method (secs)	766.593	13.796	3.468

connect-4			
Support	20.0%	30.0%	40.0%
FPClose* (secs)	27.203	8.625	4.375
CLOSET+ (secs)	29.390	8.593	4.688
Our method (secs)	10.312	3.453	2.328

Figure 8: Speed comparison between our method for mining odds ratio patterns—based on simultaneous mining of generators and closed patterns—and two state-of-the-art algorithms for mining closed patterns. The odds ratio threshold used in our method is 2.5. Note that this threshold is not required for mining frequent closed patterns where only a support threshold is required.