Structural Geography of the Space of Emerging Patterns

Jinyan Li and Limsoon Wong
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
Email: {jinyan,limsoon}@i2r.a-star.edu.sg

Abstract

Describing and capturing significant differences between two classes of data is an important data mining and classification research topic. In this paper, we use emerging patterns to describe these significant differences. Such a pattern occurs in one class of samples—its "home" class—with a high frequency but does not exist in the other class, so it can be considered as a characteristic property of its home class. We call the collection of all such patterns a space. Beyond the space, there are patterns that occur in both of the classes or that do not occur in any of the two classes. Within the space, the most general and most specific patterns bound the other patterns in a lossless convex way. We decompose the space into a terrace of pattern plateaus based on their frequency. We use the most general patterns to construct accurate classifiers. We also use these patterns in the bio-medical domain to suggest treatment plans for adjusting the expression levels of certain genes so that patients can be cured.

Keywords: space of emerging patterns, convexity, structure decomposition, medical treatment plan, PCL classifier.

1 Introduction

Patterns are often used in separating different classes of data. For example, the pattern $\{Odor = none,\ Gill\_size = broad,\ Ring\_number = 1\}$ can be used to distinguish edible and poisonous mushrooms, because all the three conditions in this pattern are satisfied by about 64% of 4208 edible mushrooms stored in a data set in [3], but none of the stored 3916 poisonous mushrooms [3] satisfies these three conditions. Although this pattern is not 100% occurring in every edible mushroom, its absence in the whole poisonous class makes this pattern a sharply contrasting feature between the poisonous and the edible class. As another example, a gene expression pattern can be used to diagnose whether a pediatric leukemia patient suffers from the main subtype T-ALL or other subtypes [14, 32]. This pattern contains expression ranges of two genes: $\{38242\_at < 899.95,\ 38147\_at < 4748.5\}$, where 38242_at and 38147_at denote the expression level of two genes. This pattern is discriminating, because about 96.4% of T-ALL patients exhibit such expression ranges on the two genes but all other subtypes express at different ranges for at least one of the two genes. We call these patterns emerging patterns [6, 16]. Let us provide below some definitions leading to the formal definition of emerging patterns.

Consider relational data sets that are described by a fixed number of features f_1, ..., f_n. Each feature f_i can take values from a domain V_i, which can be a range of numeric real values or a set of categorical values. Each record in any of the relational data sets is expected to take the form $\{f_1 = v_1,\ ...,\ f_n = v_n\}$, where $v_1 \in V_1$, ..., $v_n \in V_n$. We use the term "sample" to refer to such a record. Note that a numeric feature after discretization is considered as a feature that takes categorical values—each value v is an interval; and the domain V is a set of disjoint intervals.

An item is defined as a condition $f_i\ \theta_i\ v_i$ on a feature f_i and its value v_i, where $\theta_i \in \{=, <, \geq\}$. The "$38242\_at < 899.95$" and "$Odor = none$" above are two examples of an item, respectively for a numeric feature and a categorical feature. An example of an item for a discretized numeric feature f_i can be "$f_i = (100.0, 200.0]$", meaning $100.0 < f_i \leq 200.0$. Given a sample $S = \{f_1 = v_1,\ ...,\ f_n = v_n\}$ and an item $I = f_i\ \theta_i\ v_i'$, we write $S \models I$ to mean $v_i\ \theta_i\ v_i'$ holds in the domain V_i. We assume that in the real domain, $\{=, <, \geq\}$ have the usual meaning; and that in a categorical domain, $=$ is the only meaningful operation and has the usual meaning. We further say that S contains I, or I occurs in S, if $S \models I$.

A pattern P is defined as a set of items $\{f_{i_1}\ \theta_{i_1}\ v_{i_1},\ ...,\ f_{i_k}\ \theta_{i_k}\ v_{i_k}\}$, where f_{i_1}, ..., f_{i_k} are distinct. Such a pattern can be viewed as a conjunctive condition. We write $S \models P$ if $S \models f_{i_j}\ \theta_{i_j}\ v_{i_j}$ for each $f_{i_j}\ \theta_{i_j}\ v_{i_j} \in P$. We say a sample S contains a pattern P, or P "occurs" in S, if $S \models P$.

Given a set of samples $\mathcal{A}$ which is also called a data set, the occurrence $count^{\mathcal{A}}(P)$ of a pattern P in $\mathcal{A}$ is the number of samples in $\mathcal{A}$ that contain P, and hence

$$count^{\mathcal{A}}(P) = |\{S \in \mathcal{A} \mid S \models P\}|$$

The support $support^{\mathcal{A}}(P)$ of a pattern P in $\mathcal{A}$ is the proportion of samples in $\mathcal{A}$ that contain P, and hence

$$support^{\mathcal{A}}(P) = \frac{count^{\mathcal{A}}(P)}{|\mathcal{A}|}$$

Given two data sets $\mathcal{A}$ and $\mathcal{B}$, the growth rate $growth^{\mathcal{A}\to\mathcal{B}}(P)$ of a pattern P from $\mathcal{A}$ to $\mathcal{B}$ is defined as the ratio of the support of P in $\mathcal{B}$ to that of P in $\mathcal{A}$, and hence

$$growth^{\mathcal{A}\to\mathcal{B}}(P) = \frac{support^{\mathcal{B}}(P)}{support^{\mathcal{A}}(P)}$$

For two classes, a pattern can have a very low or even zero occurrence in one class and yet a high occurrence in the other class. Therefore, the growth rate is sometimes a finite number, and is sometimes infinite.

In this paper, our study is focused on those patterns whose growth rate is infinite. We define them as emerging patterns. More formally,

Definition 1.1 *Given two data sets $\mathcal{A}$ and $\mathcal{B}$, a pattern P is an emerging pattern from $\mathcal{A}$ to $\mathcal{B}$ if $growth^{\mathcal{A}\to\mathcal{B}}(P) = \infty$ and $S \models P$ for some $S \in \mathcal{B}$. We write $EP^{\mathcal{A}\to\mathcal{B}}$ to denote the set of all emerging patterns from $\mathcal{A}$ to $\mathcal{B}$.*

That is, an emerging pattern—EP for short—is a pattern whose occurrence in one class is non-zero but in the other class is zero. The class in which an EP has a non-zero occurrence is called the EP's home class. The other class in which the EP has zero occurrence is called the EP's counterpart class. We often use the phrase "an EP of class $\mathcal{A}$" or its equivalent to mean "an EP whose home class is $\mathcal{A}$."

Given two non-empty classes of samples, we are interested in the following problems:

- What patterns are the most general and most specific EPs? Given patterns P and P', we write $P \Rightarrow P'$ if $S \models P$ whenever $S \models P'$ for every possible sample S. The pattern P is more general than the pattern P' if $P \Rightarrow P'$. In this case, P' is also said to be more specific than P. In what follows unless specially specifies, we assume that all the features are of categorical values (including discretized values), and hence $P \Rightarrow P'$ if and only if $P \subseteq P'$ for any patterns P and P'.

- Can we derive other emerging patterns directly from the most general and most specific EPs? In other words, can the whole space be concisely represented using only the most general and most specific EPs in a lossless way?

- Is the space of emerging patterns organizable?

- How to make effective use of emerging patterns to solve real-world applications?

We organize our pursuit of these problems as follows. We begin in Section 2 with a discussion on a method that can be used to efficiently discover most general EPs. Such a method is important because the most general EPs are typically EPs that have the highest occurrence in their home class, and the discovery of these patterns is expensive. We prove that EP spaces are convex spaces. Therefore, the discovery of the most general and specific patterns can be viewed as the discovery of all the patterns since the other patterns can be directly derived from them. Furthermore, we do not need to exhaustively enumerate all EPs to discover the most general patterns.

Then, to organize the patterns in a space, we decompose in Section 3 the space into a series of sub-spaces, each called a plateau space, consisting of patterns with the same high support. After sorting these sub-spaces, we can see a terrace of pattern plateaus. Outside the space, some patterns may be interesting as well. In particular, we discuss in Section 4 patterns that are just one-step away from the bounds of the EP spaces.

After the theoretical investigations above into the structural geography of the space of emerging patterns, we turn to describe the practical use of emerging patterns in the bio-medical domain. We suggest in Section 5 a treatment plan to cure cancer disease cells based on the most general emerging patterns discovered from gene expression profiling data. In this suggestion, the emerging patterns are interpreted as common characteristics of the cancer cells or the normal cells. Let the disease characteristics disappear from a cancer cell and let the normal characteristics appear in it, then this cancer cell would be converted into a normal one, similar to the idea of using gene therapies [29, 31] to cure cancer patients. This can be done by creating a treatment plan that adjusts some genes' expression level.

Finally, we also introduce in Section 6 a new classifier, PCL, which is a shorthand for Prediction by Collective Likelihoods of emerging patterns. This classifier also makes use of the most general emerging patterns. Its performance is comparable to the best of the classical classification algorithms such as C4.5 [24], Bagging [4], Boosting [8], SVM [5], and k-nearest neighbour on a wide range of bio-medical data sets.

2 The Most General and Most Specific EPs

Suppose P is an emerging pattern, then $|P| \geq 1$. That is, an emerging pattern must contain at least one feature. The addition of any items to or the removal of any items from X, generating a proper superset or a proper subset of P respectively, may or may not result in a new EP. The reason is that a proper subset of P may occur in P's counterpart class, and a proper superset of P may not occur in P's home class again. So the space of emerging patterns can have interesting boundaries. In fact, this space satisfies convexity.

Proposition 2.1 *Given two data sets $\mathcal{A}$ and $\mathcal{B}$, the collection of all EPs with $\mathcal{A}$ as their home class form a convex space. That is, for all emerging patterns*

$X \in EP^{\mathcal{B}\to\mathcal{A}}$ and $Y \in EP^{\mathcal{B}\to\mathcal{A}}$ and for each Z such that $X \Rightarrow Z \Rightarrow Y$, it is the case that Z is also an EP of $\mathcal{A}$.

The full proof for this proposition can be found in [16] for the case of all features taking categorical values. We also claim that this proposition still holds even when not all features are of categorical values. The proof for the latter case is similar to that of the former case. We omit it here.

It is known that a convex pattern space $\mathbf{C}$ can be concisely represented by a border $\langle \mathcal{L}, \mathcal{R} \rangle$, where $\mathcal{L}$ and $\mathcal{R}$ are two subsets of patterns of $\mathbf{C}$, such that

- $\mathcal{L}$ is anti-chain, $\mathcal{R}$ is anti-chain,
- each $X \in \mathcal{L}$ is more general than some $Y \in \mathcal{R}$,
- each $Y \in \mathcal{R}$ is more specific than some $X \in \mathcal{L}$, and
- $\mathbf{C} = \{Z \mid \exists X \in \mathcal{L}, \exists Y \in \mathcal{R}, X \Rightarrow Z \Rightarrow Y\}$.

In fact, $\mathcal{L}$ are the most general patterns in $\mathbf{C}$, and $\mathcal{R}$ the most specific patterns in $\mathbf{C}$. We also write $[\mathcal{L}, \mathcal{R}]$ for $\mathbf{C}$. Note that $\langle \mathcal{L}, \mathcal{R} \rangle$ and $[\mathcal{L}, \mathcal{R}]$ are two different notions. The former is the border of $\mathbf{C}$, but the latter is $\mathbf{C}$ itself.

Therefore, the pattern space $EP^{\mathcal{B}\to\mathcal{A}}$ can be concisely represented by a border

$$\langle \mathcal{L}, \mathcal{R} \rangle$$

where patterns in $\mathcal{L}$ are the most general emerging patterns, and patterns in $\mathcal{R}$ are the most specific emerging patterns. We call all these EPs boundary EPs.

All proper subsets of any of the most general emerging patterns must occur in both classes with non-zero occurrence. So, the most general EPs are the EPs with the highest occurrences. Next, we describe how to efficiently discover the border of an EP space, especially the most general EPs, i.e. the patterns in $\mathcal{L}$.

Let $\mathcal{A}$ and $\mathcal{B}$ be two data sets described by n features and each feature be drawn from a set of i categorical values. Then there are 2^{n*i} possible patterns in $\mathcal{A}$ and $\mathcal{B}$. Hence naive methods to extract all emerging patterns and then boundary EPs would be too expensive. More efficient methods for extracting only boundary emerging patterns are therefore crucial to the practical uses of emerging patterns. Here, we discuss an efficient border-based algorithm to discover the border of an EP space when the input data sets are described solely by features of categorical values.

Proposition 2.2 *Let $\mathcal{A}$ and $\mathcal{B}$ be two data sets. Suppose $\mathcal{A}$ and $\mathcal{B}$ have no duplicate and are described by the same features of categorical values. Then $EP^{\mathcal{B}\to\mathcal{A}}$ is given by $[\{\{\}\}, \mathcal{A}] - [\{\{\}\}, \mathcal{B}]$.*

Observe that $[\{\{\}\}, \mathcal{A}]$ is the set of all patterns that have a non-zero support in $\mathcal{A}$, and $[\{\{\}\}, \mathcal{B}]$ is the set of all patterns that have a non-zero support in $\mathcal{B}$. So, the set difference produces exactly $EP^{\mathcal{B}\to\mathcal{A}}$.

Having rewritten emerging patterns to this border formulation, we can derive a more efficient approach to discovering the left boundary EPs in $\mathcal{L}$. Let

$\{A_1, \ldots, A_n\} \subseteq \mathcal{A}$ be samples of $\mathcal{A}$ that do not occur in $\mathcal{B}$. Let $\mathcal{B} = \{B_1, \ldots, B_m\}$. Then

$$\begin{aligned} & [\{\{\}\}, \mathcal{A}] - [\{\{\}\}, \mathcal{B}] \\ = \; & [\{\{\}\}, \{A_1, \ldots, A_n\}] - [\{\{\}\}, \{B_1, \ldots, B_m\}] \\ = \; & \bigcup_i^n ([\{\{\}\}, \{A_i\}] - [\{\{\}\}, \{B_1, \ldots, B_m\}]) \\ = \; & [\mathcal{L}, \{A_1, \ldots, A_n\}] \end{aligned}$$

where

$$\mathcal{L} = \bigcup_i^n (\textsc{Min}\{\{s_1, \ldots, s_m\} \mid s_j \in A_i - B_j, 1 \leq j \leq m\})$$

and $\textsc{Min}(\mathcal{S})$ denotes the collection of the most general patterns for a given pattern collection $\mathcal{S}$.

The following proposition can be iteratively used to prove the correctness of the above algorithm.

Proposition 2.3 *Let* $D = [\{\{\}\}, \{U\}] - [\{\{\}\}, \{S_1, \cdots, S_m\}]$. *Let* $\mathcal{S} = \{\{s_1,$ ..., $s_m\} \mid s_i \in U - S_i,\ 1 \leq i \leq m\}$. *Then*

1. $\mathcal{S} \subseteq D$;
2. *for each* $Y \in D$, *there is* $X \in \mathcal{S}$ *and* $X \subseteq Y$; *and*
3. $\textsc{Min}(D) = \textsc{Min}(\mathcal{S})$.

Proof. To settle Part 1 that $\mathcal{S} \subseteq D$, we proceed as follows. Let $P = \{s_1,$ ..., $s_m\}$ be in $\mathcal{S}$. By definition, for each $s_i \in P$, it is the case that $s_i \in U$. Hence, $P \in [\{\{\}\}, \{U\}]$. Also by definition, for each $s_i \in P$, it is the case that $s_i \notin S_i$. Hence, $P \notin [\{\{\}\}, \{S_1, \cdots, S_m\}]$. Therefore, $P \in D = [\{\{\}\}, \{U\}] - [\{\{\}\}, \{S_1, \ldots, S_m\}]$. Thus, $\mathcal{S} \subseteq D$.

To settle Part 2 that each $Y \in D$ is a superset of some $X \in \mathcal{S}$, we proceed as follows. Let Y be in D. Then by definition of D, $Y \not\subset S_1$, ..., $Y \not\subset S_m$. Then there are $s_1 \in Y$, ..., $s_m \in Y$ such that $s_1 \notin S_1$, ..., and $s_m \notin S_m$. Let $X = \{s_1, ..., s_m\}$. Then $X \subseteq Y$ and, by definition of S, X is in $\mathcal{S}$.

To settle Part 3, we proceed by first proving $\textsc{Min}(D) \subseteq \textsc{Min}(\mathcal{S})$ and then proving $\textsc{Min}(\mathcal{S}) \subseteq \textsc{Min}(D)$. We now prove $\textsc{Min}(D) \subseteq \textsc{Min}(\mathcal{S})$. Suppose $M_D \in \textsc{Min}(D)$. By Part 2, there exists $X \in \mathcal{S}$ such that $X \subseteq M_D$. By Part 1, $X \in D$. Since M_D is most general in D, $X = M_D$. So $M_D \in \mathcal{S}$. We now prove that $M_D \in \textsc{Min}(\mathcal{S})$. Assume $M_D \notin \textsc{Min}(\mathcal{S})$. Then there exists $Z \in \textsc{Min}(\mathcal{S})$ such that $Z \subset M_D$. By Part 1, $Z \in D$. Since $M_D \in \textsc{Min}(D)$, $Z \subset M_D$ cannot be true, a contradiction.

We next prove $\textsc{Min}(\mathcal{S}) \subseteq \textsc{Min}(D)$. Suppose $M_S \in \textsc{Min}(\mathcal{S})$. By Part 1, $M_S \in D$. Assume that $M_S \notin \textsc{Min}(D)$. Then there exists some $Y \in \textsc{Min}(D)$ such that $Y \subset M_S$. By Part 2, there exists some $Y' \in \mathcal{S}$ such that $Y' \subseteq Y$. So, $Y' \subset M_S$. But this contradicts the assumption that $M_S \in \textsc{Min}(\mathcal{S})$. Therefore $M_S \in \textsc{Min}(D)$. This completes the proof. ∎

We note that

$$\begin{aligned}\mathcal{L} &= \textsc{Min}(\bigcup_i^n \{\{s_1, \ldots, s_m\} \mid s_j \in A_i - B_j, 1 \leq j \leq m\} \\ &= \bigcup_i^n (\textsc{Min}\{\{s_1, \ldots, s_m\} \mid s_j \in A_i - B_j, 1 \leq j \leq m\})\end{aligned}$$

See the proof in [16]. So, using Proposition 2.3 iteratively, we can prove the correctness of our algorithm to discover the border of an EP space.

Example 2.4 *Consider* $[\{\{\}\}, \{\{1,2,3,4\}\}] - [\{\{\}\}, \{\{2,3\}, \{2,4\}, \{3,4\}\}]$. *For convenience of writing, we use* 1234 *as a shorthand to represent* $\{1,2,3,4\}$, *similarly for representing other sets. Let* $U = 1234$, $S_1 = 23$, $S_2 = 24$, *and* $S_3 = 34$. *Note that* $U - S_1 = 14, U - S_2 = 13, U - S_3 = 12$. *By Proposition 2.3,*

$$\begin{aligned}& \textsc{Min}([\{\{\}\}, \{1234\}] - [\{\{\}\}, \{23, 24, 34\}]) \\ = \; & \textsc{Min}(\{111, 112, 131, 132, 411, 412, 431, 432\}) \\ = \; & \textsc{Min}(\{1, 12, 13, 123, 14, 124, 134, 234\}) \\ = \; & \{1, 234\}\end{aligned}$$

Observe that 111 *is a bag (multi-set) with three occurrences of* 1. *So,* $[\{\{\}\}, \{1234\}] - [\{\{\}\}, \{23, 24, 34\}] = [\{1, 234\}, \{1234\}]$.

We can see that this algorithm is much more efficient to find the left bound of the border than a naive algorithm of enumerating all subsets of 1234, 23, 24, *and* 34.

More discussions on the efficiency and refinements of the idea above to discover the most general emerging patterns can be found in [6, 16, 33], where border-based algorithms and constraint-based algorithms are used. In particular, we note the following main points from these papers:

1. To handle features on the domain of real numeric values, there is an extra preparation step of discretizing such feature values. This discretization step can be performed using an entropy method [7].

2. To handle data of very high dimension, there is an extra preparation step of selecting top-ranked features. This feature selection step can be performed using the entropy method [7], the χ^2 method [18], *etc.* Usually, we select 20 top-ranked features as used in this paper. We do not recommend any numbers that are larger than 100 because the EP discovery algorithm would produce very large number of patterns.

3. To discover the most general emerging patterns from the (discretized) data, it is efficient to use border-based algorithms based on the idea presented above.

3 A Terrace of Pattern Plateaus: Structural Decomposition of EP Spaces

The support of an emerging pattern in its home class must be one of these values $\{1/m, 2/m, ..., m/m\}$, where m is the number of samples in this emerging

pattern's home class. Different emerging patterns may have the same occurrence in their home class. Based on this idea, we organize a space of emerging patterns and partition the space into a series of sub-spaces. First, we define plateau EPs and plateau spaces:

Definition 3.1 *Let two data sets $\mathcal{A}$ and $\mathcal{B}$ described by the same features be given. Let v be a real number such that $0 < v \leq 1$. Then $P_v^{\mathcal{B}\to\mathcal{A}} = \{P \in EP^{\mathcal{B}\to\mathcal{A}} \mid support^{\mathcal{A}}(P) = v\}$ is called a plateau space at significance level v from $\mathcal{B}$ to $\mathcal{A}$. The emerging patterns $P \in P_v^{\mathcal{B}\to\mathcal{A}}$ are called plateau EPs at significance level v from $\mathcal{B}$ to $\mathcal{A}$. If $\mathcal{A}$ and $\mathcal{B}$ are understood or are unimportant, we suppress them in our notations and simply say P_v-space. If v is understood or is unimportant, we suppress it and simply say P-space.*

All patterns in a P_v-space are at the same significance level in terms of their occurrence in both their home class and counterpart class—the occurrence in their counterpart class is zero by definition. Using P_v-spaces, we can rewrite the space of EPs from $\mathcal{B}$ to $\mathcal{A}$ as the union:

$$EP^{\mathcal{B}\to\mathcal{A}} = \bigcup_{v\in\left\{\frac{1}{|\mathcal{A}|},\frac{2}{|\mathcal{A}|},\cdots,\frac{|\mathcal{A}|}{|\mathcal{A}|}\right\}} P_v^{\mathcal{B}\to\mathcal{A}}$$

Note that

1. some P_v-spaces are empty—*i.e.*, there does not exist any emerging pattern at the significance level v; and
2. every emerging pattern belongs to one and only one of these P_v-spaces—*i.e.*, no overlapping exists between any two of these P_v-spaces.

Thus we can partition an EP space into a series of non-overlapping plateau spaces. Each plateau space matches a particular support or significance level v. These plateau spaces can be sorted into a descending order according to their significance level. Then such a structural decomposition of an EP space can be viewed as a terrace of plateau spaces.

Next we prove that all P_v-spaces satisfy the nice property of convexity. This means that a P_v-space can be succinctly represented by its most general and most specific patterns.

Theorem 3.2 *Given two classes of data $\mathcal{A}$ and $\mathcal{B}$ described by the same set of features, every non-empty $P_v^{\mathcal{B}\to\mathcal{A}}$-space is a convex space.*

Proof. By definition, a $P_v^{\mathcal{B}\to\mathcal{A}}$-space is the set of all plateau EPs with support level v in their home class $\mathcal{A}$. Suppose two patterns $X \in P_v^{\mathcal{B}\to\mathcal{A}}$ and $Z \in P_v^{\mathcal{B}\to\mathcal{A}}$ satisfy $X \Rightarrow Z$. Then to prove the convexity of $P_v^{\mathcal{B}\to\mathcal{A}}$, we need to show that $Y \in P_v^{\mathcal{B}\to\mathcal{A}}$ whenever $X \Rightarrow Y \Rightarrow Z$. We proceed by observing the followings:

1. By definition of $P_v^{\mathcal{B}\to\mathcal{A}}$, X does not occur in any sample in $\mathcal{B}$. Since $X \Rightarrow Y$, it is the case that Y does not occur in any sample in $\mathcal{B}$.

2. By definition of $P_v^{\mathcal{B}\to\mathcal{A}}$, $support^{\mathcal{A}}(Z) = v$. Since $Y \Rightarrow Z$, it is the case that $support^{\mathcal{A}}(Y) \geq v$.

3. Since $X \in P_v^{\mathcal{B}\to\mathcal{A}}$ and $Z \in P_v^{\mathcal{B}\to\mathcal{A}}$, we have $support^{\mathcal{A}}(X) = support^{\mathcal{A}}(Z) = v$. Furthermore, $support^{\mathcal{A}}(X) \geq support^{\mathcal{A}}(Y) \geq support^{\mathcal{A}}(Z)$ because $X \Rightarrow Y \Rightarrow Z$. So, $support^{\mathcal{A}}(Y) = v$.

Combining the first two points, we have Y is an EP of $\mathcal{A}$. By the third point, we conclude $Y \in P_v^{\mathcal{B}\to\mathcal{A}}$. Therefore, the $P_v^{\mathcal{B}\to\mathcal{A}}$-space is a convex space. ∎

As a plateau space $P_v^{\mathcal{B}\to\mathcal{A}}$ is a convex space, it can be concisely represented by a border consisting of two bounds. Suppose an EP space, for example $EP^{\mathcal{B}\to\mathcal{A}}$, consists of p number of non-empty plateau spaces, then

$$EP^{\mathcal{B}\to\mathcal{A}} = \bigcup_{i=1}^{p} [\mathcal{L}_i, \mathcal{R}_i]$$

where $\langle \mathcal{L}_i, \mathcal{R}_i \rangle$ $(i = 1, \cdots, p)$ is the border of the ith of the p plateau spaces. So, once the boundary EPs of the p plateau spaces are known, all other EPs in $EP^{\mathcal{B}\to\mathcal{A}}$ can be derived immediately. We call this *plateau sub-space border representation* for an EP space.

Having this representation, all patterns in an EP space enriched with their support values in their home class can be derived immediately without accessing the two data sets $\mathcal{A}$ and $\mathcal{B}$ again. This is an advantage over the one-border representation of $EP^{\mathcal{B}\to\mathcal{A}} = [\mathcal{L}, \mathcal{R}]$. This is because by the one-border representation, the support of the represented EPs is not derivable if without accessing the two data sets $\mathcal{A}$ and $\mathcal{B}$. We claim that closed patterns [22] and key patterns [2] can be used to efficiently discover the borders of plateau sub-spaces of an EP space.

Next, let's briefly discuss the relation of EP spaces with version spaces [20, 12] and disjunctive version spaces [25]. Suppose an EP space, denoted $[\mathcal{L}, \mathcal{R}]$, consists of p number of non-empty plateau spaces, denoted $[\mathcal{L}_1, \mathcal{R}_1]$, $[\mathcal{L}_2, \mathcal{R}_2]$, ..., $[\mathcal{L}_p, \mathcal{R}_p]$ in a descending order of their significance level. Then, $[\mathcal{L}_1, \mathcal{R}_1]$, the non-empty plateau space with the highest support, is a version space [20, 12] if these EPs have the full 100% support level in their home class. In this sense, our EP spaces are an important extension to the concept of version spaces. This extension is a result of relaxing the strong consistency requirement of version spaces. We found that the support level of the EPs in $[\mathcal{L}_1, \mathcal{R}_1]$ is often less than 100%. So, our extension is useful. Disjunctive version spaces [25] are also an extension to the original concept of version spaces. However, unlike us, the most general patterns and how to efficiently discover them are not discussed in the work of [25].

4 Go One Step Beyond the Bounds

When we go one step outside the most specific emerging patterns of EP spaces, there are patterns that do not have any occurrence in any of the two classes.

We are not interested in these patterns. When we go one step outside the most general emerging patterns of EP spaces, there are patterns that have non-zero support in both of the two classes. Therefore, all these proper subsets of the most general EPs have a finite support growth rate between the two classes. It is interesting to see how these subsets change their support between the two classes, and which of them change at the highest ratio. For this purpose, we define shadow patterns:

Definition 4.1 *Let P be one of the most general EPs in $EP^{\mathcal{B}\to\mathcal{A}}$. Then all immediate subsets of P are called shadow patterns of P.*

Shadow patterns can be used to measure the interestingness of the most general EPs. Given a most general EP P, if the growth rate of all its shadow patterns approach ∞, then the existence of this most general EP is reasonable. This is because the *expectedness* of P being a most general EP is large. Otherwise if the growth-rates of the shadow patterns are on average around small numbers like 1 or 2, then the pattern P is adversely interesting. This is because the expectedness of P being a most general EP is small—the existence of this most general EP is "unexpected." This conflict may reveal some new insights into the correlation of the features.

Let us discuss the expectedness of being a most general EP. The expectedness of a most general EP can be roughly estimated by examining the shadow patterns of this EP. Given a class $\mathcal{A}$ of positive samples and a class $\mathcal{B}$ of negative samples, suppose P is a most general EP of the positive class, and $|P| \geq 2$. (If $|P| = 1$, then its one and only one shadow pattern is $\{\}$. The empty set has 100% support in both classes. This is a trial case.) Denote $P = \{x_1, x_2, \cdots, x_q\}$, $q = |P|$. Rewrite P as $\{x_i\} \cup A_i$, where $A_i = P - \{x_i\}$, $1 \leq i \leq q$. So, A_i are all shadow patterns of P. By definition, A_i have non-zero occurrence in the negative class, unlike its superset X, a most general EP, whose occurrence in the negative class is zero.

Suppose $support^{\mathcal{B}}(A_i) = v_i$, $1 \leq i \leq q$, and $v_1 \geq v_2 \geq \cdots \geq v_q > 0$. So, being from a non-EP to become an EP, the shadow pattern A_1 decreases its support in $\mathcal{B}$ most, compared to other A_i's. Therefore, if $support^{\mathcal{B}}(A_1)$ is close to zero, then the expectedness of $P = A_1 \cup \{x_1\}$ being an EP becomes large. Then the existence of P as an EP is reasonable. However, if $support^{\mathcal{B}}(A_1) = v_1$ is not close to zero—say $v_1 = 20\%$—then its superset P should have a non-zero occurrence in $\mathcal{B}$ as well because P is expanded from A_1 by adding only one item. Therefore, the expectedness of P being an EP is small. If P is indeed a most general EP, this happening is therefore adversely interesting.

We next present an example to show a plateau space with the highest support level in an EP space, and also show the shadow patterns of one of the most general EPs.

Example 4.2 *Two classes, $\mathcal{N}$ and $\mathcal{D}$, consists of 22 and 40 samples respectively. The samples are described by 35 features, each feature has two categorical values. (More details about this data set are presented in the next section.) The plateau space with the highest support level in $EP^{\mathcal{D}\to\mathcal{N}}$ is interesting. Every*

patterns in this plateau space has the same support level of 77.27% in $\mathcal{N}$. The left bound—i.e., the most general EPs—of this plateau space consists of 27 EPs. They (only three presented) are

$$\mathcal{L} = \{ \{6,57,69\} \quad \{6,25,57\} \quad \vdots \quad \{25,33,37,41,43,57,59,69\} \}$$

The righ bound—i.e., the most specific EPs of this plateau space—is

$$\mathcal{R} = \{ \{6,13,25,29,32,35,41,43,45,47,57,65,68,69\} \quad \{13,25,29,32,35,37,41,43,45,47,57,59,65,68,69\} \quad \{13,25,29,33,35,37,41,43,45,47,57,59,65,68,69\} \}$$

Observe that the left boundary pattern $\{6,57,69\}$ *can be expanded, without loss of any support significance, into a right boundary pattern* {6, 13, 25, 29, 32, 35, 41, 43, 45, 47, 57, 65, 68, 69}.

The most general EP $\{6,57,69\}$ *has three shadow patterns. Their occurrence in* $\mathcal{N}$ *and in* $\mathcal{D}$ *are as follows:*

Patterns	*Occurrence in* $\mathcal{N}$	*Occurrence in* $\mathcal{D}$	*Growth*
$\{6,57,69\}$	*17*	*0*	∞
$\{57,69\}$	*20*	*9*	*2.2*
$\{6,69\}$	*19*	*2*	*9.5*
$\{6,57\}$	*17*	*1*	*17.0*

Observe that the growth rate of these three shadow patterns varies remarkablely. We can also see that the item 6 is crucial in constructing the EP $\{6,57,69\}$. *Without it, the resulting pattern* $\{57,69\}$ *has a growth rate of only 2.2. But for the other two shadow patterns which contain the item 6, their growth rates are close to the maximal finite rates.*

5 An EP-based Medical Treatment Plan

Over the last three sections, we have presented several theoretical results and some discussions about emerging patterns. In this and next two sections, we discuss how to apply the concept of emerging patterns to solve real-world problems such as medical diagnosis, treatment planning, and classification. Through the analysis on the structure of EP spaces, we have understood that the most general EPs are the most important patterns. They are the boundary between EPs and non-EPs. They distinguish EPs with high support from those with low support. Also, the most general EPs are bases to construct plateau spaces and bases to generate shadow patterns. So, in this paper, we mainly introduce the use of the most general emerging patterns.

This section explores the use of the most general patterns for medical treatment planning. This treatment planning uses the most general emerging patterns discovered from gene expression profiles of normal and cancer cells to convert cancer cells into normal ones. Our idea is similar to the idea of using gene therapy [29, 31] to cure cancer diseases. By gene therapy, viral vectors (vehicles) or non-viral vectors can ferry normal genetic material into a disease cell to replace the abnormal genes such that the expression of this gene goes to normal level.

DNA microarray gene expression profiling is a breakthrough technology in molecular biology that can simultaneously measure expression levels of thousands or even tens of thousands of genes. Through the analysis and understanding of the resulting data, many studies have discovered effective biological markers for accurate diagnosis, outcome prediction, and disease subtype classification; and have found effective mechanisms towards the re-construction of gene networks and the identification of new genes in a pathway [9, 32, 26, 10, 23, 30, 28]. Gene expression profiling data are usually represented in a relational format where every feature—*viz.* a gene—is of continuous values.

Based on the concept of emerging patterns, we have two intuitions behind an approach to treatment planning. One is that top-ranked emerging patterns in the disease class can be viewed as biological characteristics of disease cells. To convert a disease cell into a normal one, these biological characteristics must be made to disappear from this disease cell. The other intuition is that top-ranked emerging patterns in the normal cell class can be viewed as biological characteristics of normal cells. A disease cell after conversion should contain many of these characteristics of the normal class. The operation to let emerging patterns contained in a disease cell disappear and to let this disease cell contain normal emerging patterns can be conducted by modulating the gene expression levels of specific genes such that some of them are up-regulated and some are down-regulated. Though the realization and implementation of this treatment plan could be very difficult, our idea is in light of gene therapies [29, 31] as mentioned above.

Next we formulate this problem. Suppose we have a class of $\mathcal{D}$ of gene expression profiles of disease cells and a class of $\mathcal{N}$ of gene expression profiles of normal cells. First, we discover the most general emerging patterns for the normal and disease class respectively. Denote the top k emerging patterns of $\mathcal{D}$ as $EP_1^{\mathcal{D}}$, $EP_2^{\mathcal{D}}$, ..., $EP_k^{\mathcal{D}}$. Denote the top k emerging patterns of $\mathcal{N}$ as $EP_1^{\mathcal{N}}$, $EP_2^{\mathcal{N}}$, ..., $EP_k^{\mathcal{N}}$. Note that a cell sample $X \in \mathcal{D}$ usually does not contain all these $EP_i^{\mathcal{D}}$, ..., $EP_k^{\mathcal{D}}$, unless the support of all these emerging patterns is 100% in the disease class. That is, $EP_1^{\mathcal{D}}$, $EP_2^{\mathcal{D}}$, ..., $EP_k^{\mathcal{D}}$ are only common characteristics of the class; any specific disease cells may contain only some of them.

In other words, given a sample gene expression profile $T = \{t_1, ..., t_n\} \in \mathcal{D}$, the problem is how to change t_1, ..., t_n, such that the resulting T' satisfies

1. $T' \not\models EP_i^{\mathcal{D}}$ for each $i = 1, ..., k$; and
2. $T' \models EP_i^{\mathcal{N}}$ for some $i = 1, ..., k$.

We use a colon tumor gene expression profiling data set [1] to demonstrate how to implement this treatment plan. This colon tumor dataset is described by 2000 genes (features), consisting of 22 normal samples and 40 colon tumor samples. Using the entropy measure [7] to discretize the data, only 135 of the 2000 genes are partitioned, and each of them has 2 intervals. The remaining 1865 genes are considered unimportant and eliminated by the method. We further concentrate on the 35 genes with the lowest entropy measure amongst the 135 genes for an easy platform where a small number of good diagnostic indicators are concentrated. So, the reduced dataset have only 35 features, each of them has two discretized values. Therefore, there are 70 items in this data set in total, as listed in Figure 1. Each item refers to an expression interval of the corresponding gene. For example, the first item means M26338 < 59.83 and the second item means M26383 ≥ 59.83, and so on. Here, M26383 is a gene name.

Next, we use the efficient border-based algorithm [6, 16] to discover the most general emerging patterns. The emerging patterns are thus combinations of intervals of gene expression levels of these relevant genes. A total of 10548 emerging patterns are found, 9540 emerging patterns for the normal class and 1008 emerging patterns for the tumour class. The top several tens of the normal class emerging patterns contain about 8 genes each and can reach a support of 77.27%, while many tumour class emerging patterns can reach a support of around 65%. Some top-ranked emerging patterns are presented in Figure 2. Note that the numbers in the emerging patterns, such as 2 or 10 in $\{2, 10\}$ of Figure 2, refer to an item in Figure 1. Hence, $\{2, 10\}$ denotes the pattern $\{\text{M26383} \geq 59.83, \text{H08393} \geq 84.87\}$.

We use a cancer cell (T1) of the colon tumor dataset as an example to show how a tumor cell is converted into a normal one. Recall the first emerging pattern $\{25, 33, 37, 41, 43, 57, 59, 69\}$ in Figure 2 is a common property of the normal cells. The eight genes involved in this emerging pattern are M16937, H51015, R10066, T57619, R84411, T47377, X53586, and U09587. Let us list the expression profile of these eight genes in T1:

genes	expression levels in T1
M16937	369.92
H51015	137.39
R10066	354.97
T57619	1926.39
R84411	798.28
T47377	662.06
X53586	136.09
U09587	672.20

However, 77.27%—17 out of 22 cases—of the normal cells have the following expression intervals for these 8 genes:

Our list	accession number	cutting points	Name
1,2	M26383	59.83	monocyte-derived neutrophil-activating ...
3,4	M63391	1696.22	Human desmin gene
5,6	R87126	379.38	myosin heavy chain, nonmuscle (Gallus gallus)
7,8	M76378	842.30	Human cysteine-rich protein (CRP) gene, ...
9,10	H08393	84.87	Collagen alpha 2(XI) chain (Homo sapiens)
11,12	X12671	229.99	heterogeneous nuclear ribonucleoprotein core ...
13,14	R36977	274.96	P03001 TRANSCRIPTION FACTOR IIIA
15,16	J02854	735.80	Myosin regulatory light chain 2, smooth muscle ...
17,18	M22382	447.04	Mitochondrial matrix protein P1 precursor
19,20	J05032	88.90	Human aspartyl-tRNA synthetase alpha-2 subunit
21,22	M76378	1048.37	cysteine-rich protein (CRP) gene, exons 5 and 6
23,24	M76378	1136.74	cysteine-rich protein (CRP) gene, exons 5 and 6
25,26	M16937	390.44	Human homeo box c1 protein mRNA
27,28	H40095	400.03	Macrophage migration inhibitory factor (Human)
29,30	U30825	288.99	Human splicing factor SRp30c mRNA
31,32	H43887	334.01	Complement Factor D Precursor
33,34	H51015	84.19	Proto-oncogene DBL Precursor
35,36	X57206	417.30	1D-myo-inositol-trisphosphate 3-kinase B isoenzyme
37,38	R10066	494.17	PROHIBITIN (Homo sapiens)
39,40	T96873	75.42	Hypothetical protein in TRPE 3'region ...
41,42	T57619	2597.85	40S ribosomal protein S6 (Nicotiana tabacum)
43,44	R84411	735.57	Small nuclear ribonucleoprotein assoc. protein ...
45,46	U21090	232.74	Human DNA polymerase delta small subunit ...
47,48	U32519	87.58	Human GAP SH3 binding protein mRNA
49,50	T71025	1695.98	Human (HUMAN)
51,52	T92451	845.7	Tropomyosin, fibroblast and epithelial muscle-type
53,54	U09564	120.38	Human serine kinase mRNA
55,56	H40560	913.77	THIOREDOXIN (HUMAN)
57,58	T47377	629.44	S-100P PROTEIN (HUMAN)
59,60	X53586	121.91	Human mRNA for integrin alpha 6
61,62	U25138	186.19	Human MaxiK potassium channel beta ...
63,64	T60155	1798.65	ACTIN, AORTIC SMOOTH MUSCLE (HUMAN)
65,66	H55758	1453.15	ALPHA ENOLASE (HUMAN)
67,68	Z50753	196.12	H.sapiens mRNA for GCAP-II/uroguanylin ...
69,70	U09587	486.17	Human glycyl-tRNA synthetase mRNA

Figure 1: The 35 top-ranked genes by the entropy measure. The index numbers in the first column are used to refer to the two expression intervals of the corresponding genes. For example, the index 1 means M26338 < 59.83 and the index 2 means M26383 ≥ 59.83. Here 59.83 is the cutting point for the expression level of this gene.

Emerging patterns	Count & Support (%) in normal tissues	Count & Support (%) in cancer tissues
$\{25, 33, 37, 41, 43, 57, 59, 69\}$	17(77.27%)	0
$\{25, 33, 37, 41, 43, 47, 57, 69\}$	17(77.27%)	0
$\{29, 33, 35, 37, 41, 43, 57, 69\}$	17(77.27%)	0
$\{29, 33, 37, 41, 43, 47, 57, 69\}$	17(77.27%)	0
$\{29, 33, 37, 41, 43, 57, 59, 69\}$	17(77.27%)	0
$\{2, 10\}$	0	28 (70.00%)
$\{10, 61\}$	0	27 (67.50%)
$\{10, 20\}$	0	27 (67.50%)
$\{3, 10\}$	0	27 (67.50%)
$\{10, 21\}$	0	27 (67.50%)

Figure 2: The top 5 emerging patterns from the normal class and the top 5 emerging patterns from the disease class, in a descending order respectively, sorted by their support in their home class.

genes	expression interval
M16937	<390.44
H51015	<84.19
R10066	<494.17
T57619	<2597.85
R84411	<735.57
T47377	<629.44
X53586	<121.91
U09587	<486.17

Comparing T1's gene expression levels with the intervals of the normal cells, we see that 5 of the 8 genes—H51015, R84411, T47377, X53586, and U09587—of the cancer cell T1 behave in a different way from those the 22 normal cells commonly express. However, the remaining 3 genes of T1 are in the same expression range as most of the normal cells. So, if the 5 genes of T1 can be down regulated to scale below those cutting points, then this adjusted cancer cell will have a common property of the normal cells. This is because $\{25, 33, 37, 41, 43, 57, 59, 69\}$ is an emerging pattern which does not occur in the cancer cells. This idea is at the core of our suggestion for this treatment plan.

Interestingly, the expression change of the 5 genes in T1 leads to a chain of other changes. These include the change that 9 extra top-ten EPs of normal cells are contained in the adjusted T1. So all top-ten EPs of normal cells are contained in T1 if the 5 genes' expression level are adjusted. As the average number of top-ten EPs contained in a normal cell is 7, the changed T1 cell will now be considered as a cell that has the most important characteristics of normal cells. So far we have adjusted only 5 genes' expression level.

Note that in this example, we set the parameter k as 10. That is, any of the top 10 EPs from the disease class should disappear from a disease cell, and

some of the top 10 EPs from the normal class should be contained in a disease cell after adjustments.

The subsequent step is to eliminate those common properties of the cancer cells that are contained in T1. By adjusting the expression level of two other genes, M26383 and H08393, the top-ten EPs of the cancer cells all disappear from T1. According to our colon tumor dataset, the average number of top-ten EPs of the cancer cells contained in a cancer cell is 6. Therefore, T1 is converted into a normal cell as it is now holding the common properties of the normal cells and does not include the common properties of the cancer cells.

By this method, all the other 39 cancer cells can be converted into normal ones after adjusting the expression levels of 10 genes or so, possibly different genes from person to person. We conjecture that this personalized treatment plan is effective if the expression of some particular genes can be modulated by suitable means.

We discuss a validation of this idea. The "adjustments" we made to the 40 colon tumour cells were based on the emerging patterns in the manner described above. If these adjustments had indeed converted the colon tumour cells into normal cells, then any good classifier that could distinguish normal vs colon tumour cells on the basis of gene expression profiles would classify our adjusted cells as normal cells. So, we established a SVM model using the original entire 22 normal plus 40 cancer cells as training data. The code for constructing this SVM model is available at `http://www.cs.waikato.ac.nz/ml/weka`. The prediction result is that all of the adjusted cells were predicted as normal cells. Although our "therapy" was not applied to the real treatment of a patient, the prediction result by the SVM model partially demonstrates the potential biological significance of our proposal. The same process can be directly applied to other types of tumors.

6 PCL: Prediction by Collective Likelihoods of Emerging Patterns

We have seen the usefulness of the most general emerging patterns in the treatment plan. In this section, we introduce a classifier that also uses the most general emerging patterns as its basis. The classifier is named PCL, which is originally proposed in our previous works [17, 14]. In this paper, we revise the classification scores so that PCL is more suitable for assessing the confidence of decision as discussed in [15]. We also report a comprehensive performance comparison between the PCL and C4.5 classifiers using both low-dimensional and high-dimensional bio-medical data sets. We begin with a description of PCL, followed by the description of the data sets and the performance report.

6.1 PCL

Given two training datasets $\mathcal{A}$ and $\mathcal{B}$ and a test sample T, the first phase of the PCL classifier is to discover the most general EPs of $\mathcal{A}$ and $\mathcal{B}$. Denote the

most general EPs of $\mathcal{A}$ as $EP_1^{\mathcal{A}}$, $EP_2^{\mathcal{A}}$, ..., $EP_i^{\mathcal{A}}$, in descending order of their support. Similarly, denote the most general EPs of $\mathcal{B}$ as $EP_1^{\mathcal{B}}$, $EP_2^{\mathcal{B}}$, ..., $EP_j^{\mathcal{B}}$, also in descending order of their support.

Suppose the test sample T contains the following EPs of $\mathcal{A}$:

$$EP_{i_1}^{\mathcal{A}}, EP_{i_2}^{\mathcal{A}}, \cdots, EP_{i_x}^{\mathcal{A}},$$

where $i_1 < i_2 < \cdots < i_x \leq i$, and the following EPs of $\mathcal{B}$:

$$EP_{j_1}^{\mathcal{B}}, EP_{j_2}^{\mathcal{B}}, \cdots, EP_{j_y}^{\mathcal{B}},$$

where $j_1 < j_2 < \cdots < j_y \leq j$.

The next step is to calculate two scores for predicting the class label of T. Suppose we use k ($k \ll i$ and $k \ll j$) top-ranked EPs of $\mathcal{A}$ and $\mathcal{B}$. Then we define the score of T in the $\mathcal{A}$ class as

$$score^{\mathcal{A}}(T) = \sum_{m=1}^{k} \frac{support^{\mathcal{A}}(EP_{i_m}^{\mathcal{A}})}{support^{\mathcal{A}}(EP_m^{\mathcal{A}})} \Bigg/ k$$

and similarly the score in the $\mathcal{B}$ class as

$$score^{\mathcal{B}}(T) = \sum_{m=1}^{k} \frac{support^{\mathcal{B}}(EP_{j_m}^{N})}{support^{\mathcal{B}}(EP_m^{N})} \Bigg/ k$$

If $score^{\mathcal{A}}(T) > score^{\mathcal{B}}(T)$, then T is predicted as the class of $\mathcal{A}$. Otherwise, it is predicted as the class of $\mathcal{B}$.

Next we demonstrate how the classification scores are computed. Suppose $k = 5$, and the support of the 5 top-ranked EPs of the class $\mathcal{A}$ are sorted as 90% ($EP_1^{\mathcal{A}}$), 85% ($EP_2^{\mathcal{A}}$), 80% ($EP_3^{\mathcal{A}}$), 75% ($EP_4^{\mathcal{A}}$), and 70% ($EP_5^{\mathcal{A}}$). Assume the test sample T contains $EP_1^{\mathcal{A}}$ (90%), $EP_3^{\mathcal{A}}$ (80%), $EP_5^{\mathcal{A}}$ (70%), $EP_7^{\mathcal{A}}$ (40%), and $EP_9^{\mathcal{A}}$ (35%). Then

$$score^{\mathcal{A}}(T) = \left(\frac{90}{90} + \frac{80}{85} + \frac{70}{80} + \frac{40}{75} + \frac{35}{70}\right) \Bigg/ 5 = 0.75$$

Note that

$$0 \leq \frac{support^{\mathcal{A}}(EP_{j_m}^{\mathcal{A}})}{support^{\mathcal{A}}(EP_m^{\mathcal{A}})} \leq 1, \quad 0 \leq score^{\mathcal{A}}(T) \leq 1,$$

$$0 \leq \frac{support^{\mathcal{B}}(EP_{j_m}^{\mathcal{B}})}{support^{\mathcal{B}}(EP_m^{\mathcal{B}})} \leq 1, \quad 0 \leq score^{\mathcal{B}}(T) \leq 1$$

All these values or scores appear to be like likelihood. This is a reason we use "collective likelihood" to name our classifier. These likelihoods can be understood as the probability of how many significant emerging patterns are contained in a test sample.

Let us explain when $score^{\mathcal{A}}(T) = 1$ and when $score^{\mathcal{B}}(T)$ is 0. The score $score^{\mathcal{A}}(T) = 1$ if and only if the test sample T satisfies all k top-ranked EPs in $\mathcal{A}$: $EP_1^{\mathcal{A}}$, $EP_2^{\mathcal{A}}$, ..., $EP_k^{\mathcal{A}}$. The other score $score^{\mathcal{B}}(T) = 0$ if and only if the test sample does not satisfy anyone of the k top-ranked EPs in $\mathcal{B}$. When such scores occur, the prediction is highly confident. If the two scores are close to each other, then the prediction should be taken carefully. But the tie-score cases rarely occur in our analyses.

PCL can be extended to apply to applications with multiple classes. Suppose we have C ($C > 2$) classes of data, denoted $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_C$. In the first phase, PCL discovers C groups of most general EPs. The cth ($1 \leq c \leq C$) group is for $\mathcal{D}_c$ (versus $\cup_{i \neq c} \mathcal{D}_i$). (The feature selection and discretization can be done as the same as dealing with typical two-class data.) Denote the ranked EPs of $\mathcal{D}_c$ as,

$$EP_1^{(c)}, EP_2^{(c)}, \cdots, EP_{i_c}^{(c)},$$

in a descending order of their support.

Suppose a test sample T contain the following EPs of $\mathcal{D}_c$:

$$EP_{j_1}^{(c)}, EP_{j_2}^{(c)}, \cdots, EP_{j_x}^{(c)},$$

where $j_1 < j_2 < \cdots < j_x \leq i_c$. The next step is to calculate c scores for predicting the class label of T. Suppose we use k ($k \ll i_c$) top-ranked EPs. Then the score of T in the $\mathcal{D}_c$ class is defined as

$$score^{\mathcal{D}_c}(T) = \sum_{m=1}^{k} \frac{support^{\mathcal{D}_c}(EP_{j_m}^{(c)})}{support^{\mathcal{D}_c}(EP_m^{(c)})} \Bigg/ k.$$

The class with the highest score is predicted as the class of T. We use the sizes of $\mathcal{D}_c$, $1 \leq c \leq C$, to break a tie.

6.2 Data Sets Description

We use two groups of bio-medical data sets to compare the performance of C4.5 (single, Bagging and Boosting) and our PCL classifier. One group includes traditional clinical data sets stored at the widely used UCI machine learning repository [3]. The other group includes recently published high-dimensional profiling data sets such as gene expression profiles and proteomic mass/charge profiles.

We use Figure 3 to summarize the background information of the 10 bio-medical data sets from the UCI machine learning repository.

The second group of data sets include 3 high-dimensional data sets for cancer diagnosis using gene expression or proteomic profiling data. Basically, all of these application are classical supervised learning problems. For example, in the pediatric leukemia data set [32], the goal is to correctly classify subtypes of this heterogeneous disease; in the ovarian tumor data set [23], it is aimed to classify tumor and normal cells for diagnostic purpose; while in the lung cancer data set [10], it is aimed to differentiate two types of disease.

Data sets	# of features	Class names	# of samples
Breast-w	9	2, 4	$699(= 458 + 241)$
Cleve	13	1, 2	$303(= 165 + 148)$
Heart	13	1, 2	$270(= 150 + 120)$
Hepatitis	19	1, 2	$155(= 32 + 123)$
HIV	8	0, 1	$362(= 248 + 114)$
Hypothyroid	29	h, n	$3163(= 151 + 3012)$
Lymph	18	2, 3	$142(= 81 + 61)$
Promoter	60	+, -	$106(= 53 + 53)$
Sick	29	sick, negative	$3772(= 231 + 3541)$
Splice	60	EI, IE, N	$3175(= 762 + 765 + 1648)$

Figure 3: Ten classical bio-medical data sets from the UCI machine learning repository. The total number of samples in a data set and the number of samples in each class are shown in the fourth column; while the third column can be used to match the data volume in a specific class.

We use Figure 4 to summarize the background information of 6 data sets for the subtype classification of the childhood leukemia disease. All these data are available at our Kent Ridge Bio-medical Data Sets Repository, its URL is `http://sdmc.i2r.a-star.edu.sg/rp/`.

The background information of the ovarian disease and the lung cancer disease data sets are summarized in Figure 5. The two data sets are also available at our website mentioned above.

Data sets	# of features	Class names	Training size	Test size
BCR-ABL	12558	BCR-ABL, others	$9 + 206$	$6 + 106$
E2A-PBX1	12558	E2A-PBX1, others	$18 + 197$	$9 + 103$
HyperL50	12558	HyperL50, others	$42 + 173$	$22 + 90$
MLL	12558	MLL, others	$14 + 201$	$6 + 106$
T-ALL	12558	T-ALL, others	$28 + 187$	$15 + 97$
TEL-AML1	12558	TEL-AML1, others	$52 + 163$	$27 + 85$

Figure 4: Data sets for the subtype classification of the childhood leukemia disease. The class names listed in the third column can be used to match the number of training or test samples in a specific class.

Data sets	# of features	Class names	Training size	Test size
Ovarian disease	15154	Cancer, Normal	162 + 91	—
Lung cancer	12533	MPM, ADCA	16 + 16	15 + 134

Figure 5: Basic information of the ovarian disease data set and the lung cancer data set. The "—" sign represents no independent test data is available.

6.3 Comparison between C4.5 and PCL

We report the accuracy and error numbers of the learning algorithms on the considered bio-medical data sets. The accuracy of a classifier is defined as the percentage of samples in a data set that are correctly classified by a classifier in a stratified 10-fold cross validation or in a validation on independent test data. The error number of a classifier is defined as the number of samples in a data set that are wrongly classified by a classifier in a stratified 10-fold cross validation or in a validation on independent test data. The latter is specially called *test* error numbers, which is widely used in the bio-medical field. When the error numbers are represented in the format $z(x : y)$, it means that x number of samples from the first class and y number of samples from the second class are misclassified, and that a total $z(= x+y)$ number samples are wrongly classified.

Our computer is a PC of DELL dimension 4100 running RedHat Linux 7.1 with a CPU speed of 886MHz and with a 512KB Ram. The main software package used in the experiments is *Weka* version 3.2, its Java open source codes are available at `http://www.cs.waikato.ac.nz/~ml/weka/` under the GNU General Public License. Our in-house softwares like PCL are coded by C++. The C4.5 (single tree, Bagging and Boosting), SVM, and k-NN programs were run under all default settings in the Weka package except that "the number of nearest neighbors to use in prediction" was reset as 3 for k-NN—the default is 1. For our PCL classifier, we set k as 5 when applied to the UCI data sets, and set k as 20 when applied to the high-dimensional bio-medical data sets.

Figure 6 summarizes the performance of the classifiers on the 10 UCI bio-medical data sets. For a simple comparison, we give the following statistics:

- Comparing PCL, C4.5, Bagging and Boosting, PCL won the best accuracy on 5 data sets—*viz.* breast-w, cleve, heart, HIV, and promoter; Bagging won on 1 data set—hypothyroid; and Boosting won the best accuracy on 4 data sets—*viz.* hepatitis, lymph, sick, and splice.

- Comparing between PCL and C4.5, PCL won on 8 data sets, while C4.5 won on the rest 2 data sets.

- Comparing between PCL and Bagging, PCL won on 6 data sets, while Bagging won on 4 data sets.

Data sets	Accuracy (%)				Error numbers			
	PCL	C4.5	Bagging	Boost	PCL	C4.5	Bagging	Boosting
Breast-w	**96.6**	94.8	96.3	96.0	24(15:9)	36(24:12)	26(14:12)	28(18:10)
Cleve	**81.8**	76.8	79.8	78.5	55(31:24)	70(32:38)	61(28:33)	65(33:32)
Heart	**83.3**	81.0	79.3	80.0	45(25:20)	50(21:29)	56(33:23)	54(25:29)
Hepatitis	80.0	78.7	80.6	**83.8**	31(7:24)	33(20:13)	30(19:11)	25(14:11)
HIV	**91.1**	85.9	85.1	89.5	32(14:18)	51(28:23)	54(30:24)	38(17:21)
Hypo	98.9	**99.2**	**99.2**	98.8	38(13:25)	25(14:11)	24(13:11)	39(17:22)
Lymph	83.0	77.5	81.7	**85.2**	24(13:11)	32(13:19)	26(8:18)	21(7:14)
Promoter	**91.5**	79.2	82.1	89.6	9(4:5)	22(10:12)	19(11:8)	11(1:10)
Sick	98.4	98.6	98.9	**99.2**	62(32:30)	52(30:22)	42(31:11)	30(20:10)
Splice	94.4	94.3	94.5	**94.7**	179 (44:42:93)	182 (23:61:98)	174 (26:54:94)	167 (27:55:85)

Figure 6: The performance of PCL and the C4.5 family algorithms on the 10 UCI bio-medical data sets. The accuracy is used for measuring the quality of both the learning algorithm and the data; while the error number is used to show exact number of mistakes made in total and in each class on a data set by a learning algorithm.

- Comparing between PCL and Boosting, PCL won on 6 data sets, while Boosting won on 4 data sets.

On a closer examination, we found that:

1. on the lymph and splice data sets where Boosting has the best accuracy, PCL has a very comparable accuracy to Boosting; and
2. on the hypothyroid data set where Bagging has the best accuracy, PCL is better than Boosting.

So, generally speaking, our PCL classifier is more accurate than the traditional decision-tree based single classifier like C4.5 or committee classifiers like Bagging and Boosting. We also find that the accuracy provided by the committee classifiers are all better than C4.5. These results confirm that the use of multiple significant rules is an effective way to improve the accuracy of C4.5.

We next report experimental results on the 8 high-mensional profiling data sets. The results are summarized in Figure 7 and Figure 8. We can see that PCL is consistently—with only one exception on the TEL-AML1 data set—better than or equal to the performance of the C4.5 family of algorithms.

Let us explain a bit more about the results on the lung cancer data set. The training part of this data set is small, having only 32 samples, but the test data consists about 4 times more samples than the training size. The C4.5 tree derived from this training data is very simple. It uses only one feature to 100% accurately classify the 32 samples. However, this thin-structure tree makes 27 mistakes on the test data. The Boosting algorithm makes the same

Datasets	Error Numbers for Test Data					
	PCL	C4.5	Bagging	Boosting	SVM	3-NN
BCR-ABL	**1:0**	1:4	2:0	1:4	1:1	**1:0**
E2A-PBX1	0:0	0:0	0:0	0:0	0:0	0:0
HyperL50	2:2	4:5	4:2	1:4	**0:3**	1:4
MLL	**0:0**	1:1	**0:0**	1:1	**0:0**	**0:0**
T-ALL	**0:0**	0:1	0:1	0:1	**0:0**	**0:0**
TEL-AML1	2:0	3:1	**1:0**	**1:0**	1:1	2:0
Total errors	7	21	10	14	7	8

Figure 7: The error numbers of 6 classification algorithms on the data sets for the subtype classification of childhood leukemia.

Datasets	Error Numbers for Test Data					
	PCL	C4.5	Bagging	Boosting	SVM	3-NN
Ovarian	**4(3:1)**	10(5:5)	8(4:4)	5(3:2)	5(2:3)	5(2:3)
Lung Cancer	3(1:2)	27(4:23)	18(2:16)	27(4:23)	**1(1:0)**	**1(1:0)**

Figure 8: The error numbers of 6 classification algorithms on the ovarian disease data set and the lung cancer data set.

number of mistakes as C4.5 makes. This is because the Boosting committee is a singleton—only one tree is contained—and is thus unable to take advantage of the power of a real committee. This indicates that C4.5 has well learned only one aspect of the training data, and has ignored many other significant rules. So the possibility of making mistakes increases.

However, PCL discovers a total of 39 significant emerging patterns, and it uses them as a committee. So, it is not a surprise to see that PCL is much better than C4.5 on this data set.

Compared to the non-linear classifiers such as SVM and nearest-neighbour, the performance of PCL is comparable and sometime is better. See Figures 7 and 8. PCL's advantage over SVM and nearest neighbour is that PCL provides easily understandable patterns and rules.

We also conduct experiments to see the speed differences between PCL and C4.5. C4.5 is faster than PCL. This is because C4.5 is a heuristic search method, while PCL is a global search method. In theoretical worst cases, the candidate patterns searched by PCL is exponential to the number of features in a data set. This is a reason why PCL needs to select top-ranked features when handling high-dimensional medical data. However, on all the data sets presented in this paper, PCL completes each of the experiments within a couple of minutes, spending seconds on small data sets and longer time on other data sets. These timing figures are not that long for diagnostic purposes.

7 Related Work

The concept of emerging patterns was originally proposed in [6]. Under that definition, emerging patterns include patterns with both infinite and finite growth rates (larger than a threshold). For example, if a pattern has a support of 10% in one class, but in the other class its support is 2%. It is an emerging pattern with a growth rate of 5. So, the emerging patterns studied in this paper are a special type of emerging patterns under the original definition. We have shown that all EP spaces of this study satisfy the nice property of convexity. However, emerging patterns under the original definition do not satisfy this property. Therefore, the discovery, representation, and application of those emerging patterns become much more difficult.

Version spaces [20] and disjunctive version spaces [25] are concepts that are closely related to EP spaces. A version space was originally defined as a set of hypotheses—equivalent to patterns in this work—in which every hypothesis must be contained in (or match) all positive instances but no negative instances. So, an EP space is an extension to a version space by relaxing the strict consistency conditions required in the version space. Unlike our border-structure representation, a disjunctive version space [25] is represented by a disjunction of conjunctions of many scoped disjunctive hypotheses. Only conceptually, a disjunctive version space is an implicit representation of all EPs, but the most general EPs and algorithms to discover them are not discussed in disjunctive version spaces. In a different way, an EP space—more precisely, the border of the space—is an explicit description of the most general and specific EPs, in addition to being an implicit representation of all other EPs. Such a concise easily understandable semi-explicit representation can meet the demand of efficiently retrieving solid important EPs, as well as avoid the exhaustive enumeration of all the patterns in the space. We summarize these differences using a list of different representations of EP spaces.

$$\begin{aligned} EP^{\mathcal{B}\to\mathcal{A}} &= [\{\{\}\},\mathcal{A}] - [\{\{\}\},\mathcal{B}] \\ &= \vee_{Ex\in\mathcal{A}}(\wedge_{Ce\in\mathcal{B}} D(Ex,Ce)) \\ &= \langle\mathcal{L},\mathcal{R}\rangle \\ &= \{EP_1, EP_2, \cdots, EP_x\} \end{aligned}$$

where $[\{\{\}\},\mathcal{A}] - [\{\{\}\},\mathcal{B}]$ is the rewritten form of an EP space as discussed in Proposition 2.2. The second representation is disjunctive version space. The fourth representation is an explicit exhaustive enumeration of all individual EPs. The third one, constructed by the most general and most specific EPs, is a concise border description of the fourth representation. Explicit EPs do appear in these two representations. However, the first and second representations do not provide any explicit EPs though each of them conceptually covers all EPs. Our algorithms begin with the first representation, moving to derive the second one, and finally terminating at the third. With simple extra computation, the border structure can be transformed into an exhaustive enumeration of all the EPs. We note that these representations may have their own advantages in

different situations. From the data mining perspective, we believe that the third one is optimal as it directly informs us which patterns are the most general EPs, which are the most specific, and where the other EPs reside.

Other related work to version spaces can be found in [21, 12, 27, 13], where generalization of version spaces are discussed. All of them are different from ours. Though Gunter *et al.* [11] are concerned with efficiency issues of assumption-based truth maintenance systems, their work contains some ideas similar to ours, including the representation of interval closed collections using borders. The work [19] proposes positive and negative borders to estimate the efficiency of a level-wise algorithm which can be used to discover interesting rules such as association rules, strong rules, and frequent episodes, but not including emerging patterns.

The structure decomposition of EP spaces into a terrace of plateau subspaces is first proposed in this paper, though a draft concept of plateau patterns and spaces were roughly studied in our earlier work [17] where plateau spaces are defined differently. Plateau patterns defined in [17] are those generated only from boundary EPs that have the same home frequency. As such, the EP space's complementary set to these plateau spaces is not a plateau space. Therefore, EP space cannot be decomposed to a terrace of plateau spaces.

The PCL classifier is inspired and motivated by our previous analysis on gene expression profiling data in our bioinformatics research [17, 14, 32]. The PCL classifier has achieved a milestone result for the subtype classification of the heterogeneous disease of pediatric leukemia. So far, it is the most accurate classifier on this application. In this paper, we revise the method to calculate the classification scores so that the scores are more easily interpreted. In addition to this, we report for the first time a comprehensive comparison between the performance of PCL and C4.5 on a wide range of bio-medical data sets. PCL is a rule-based classifier. Its easy interpretation from raw data to knowledge patterns, together with its high accuracy, makes this classifier important in the field of machine learning.

8 Conclusion

In this paper, we have studied the structure of EP spaces. EP spaces are bounded convex spaces—the most general and most specific emerging patterns bound the space in a lossless way. EP spaces are organizable—every EP space can be decomposed into a terrace of plateau sub-spaces. Having this structure, we then have a clearer view of where version spaces are located in EP spaces. Shadow patterns are also interesting as such a neighbourhood of EP spaces can be used to judge the interestingness of most general EPs.

The proposed medical treatment plan and the PCL classifier have demonstrated the usefulness of the most general emerging patterns for real applications. The significance of the PCL classifier is not only in its accuracy, more importantly, is in the easy interpretation of the patterns and rules.

As future research work, we are going to pursue along the following three

directions:

- This paper is focused on the mining of all the most general emerging patterns. But, the proposed PCL classifier makes use of only top k of them. So, the question is how to efficiently pinpoint the top k instead of the whole left bound of an EP space.
- For the PCL classifier, we give one way of defining the score of a sample T relative to two data sets. Some other ideas are possible. For example, one way is to sum the supports of EP_{i_m} and divide it by the sum of the supports of EP_m. Basically, it is a question of how to integrate the discriminating power of emerging patterns for classification.
- For the treatment plan that converts tumor cells into healthy cells, our heuristic is to modulate the expression levels of some specific genes according to top-ranked emerging patterns. In general, it seems like a complex optimization problem to select the appropriate set of genes to be modified for the treatment. Are there any in-silico ideas underlying the biological nature of the application? We expect some biological guidance such as those from gene therapies [29, 31].

Acknowledgement

Thanks to Guozhu Dong and Kotagiri Ramamohanarao for their supervision of the first author's PhD thesis. Some results in this paper appeared in this PhD thesis.

References

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of National Academy of Sciences of the United States of American*, 96:6745–6750, 1999.

[2] Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2):66–75, 2000.

[3] C.L. Blake and P.M. Murphy. The UCI machine learning repository. [`http://www.cs.uci.edu/~mlearn/MLRepository.html`]. In *Irvine, CA: University of California, Department of Information and Computer Science*, 1998.

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[5] C.J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[6] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In Surajit Chaudhuri and David Madigan, editors, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 43–52, San Diego, CA, 1999. ACM Press.

[7] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In Ruzena Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1029. Morgan Kaufmann, 1993.

[8] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, Bari, Italy, July 1996. Morgan Kaufmann.

[9] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J.R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, October 1999.

[10] Gavin J. Gordon, Roderick V. Jensen, Li-Li Hsiao, Steven R. Gullans, Joshua E. Blumenstock, Sridhar Ramaswamy, William G. Richards, David J. Sugarbaker, and Raphael Bueno. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62:4963–4967, 2002.

[11] Carl A. Gunter, Teow-Hin Ngair, and Devika Subramanian. The common order-theoretic structure of version spaces and ATMS's. *Artificial Intelligence*, 95:357–407, 1997.

[12] H. Hirsh. Generalizing version spaces. *Machine Learning*, 17:5–46, 1994.

[13] Sau Dan Lee and Luc De Raedt. An algebra for inductive query evaluation. In *ICDM*, pages 147–154, 2003.

[14] Jinyan Li, Huiqing Liu, James R. Downing, Allen Eng-Juh Yeoh, and Limsoon Wong. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. *Bioinformatics*, 19:71 –78, 2003.

[15] Jinyan Li and Hwee-Leng Ong. Feature space transformation and decision results interpretation (the best paper of the conference). In *Proceedings of the First Asia-Pacific Bioinformatics Conference*, pages 129 – 137, Adelaide, Australia, 2003. Australian Computer Society.

[16] Jinyan Li, Kotagiri Ramamohanarao, and Guozhu Dong. The space of jumping emerging patterns and its incremental maintenance algorithms.

In *Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, USA*, pages 551–558, San Francisco, June 2000. Morgan Kaufmann.

[17] Jinyan Li and Limsoon Wong. Geography of differences between two classes of data. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2002*, pages 325 – 337, Helsinki, Finland, 2002. Springer-Verlag.

[18] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the IEEE 7th International Conference on Tools with Artificial Intelligence*, pages 338 – 391, 1995.

[19] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1:241–258, 1997.

[20] T.M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[21] Steven W. Norton and Haym Hirsh. Learning DNF via probabilistic evidence combination. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 220–227. Morgan Kaufmann, 1993.

[22] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory (ICDT)*, pages 398–416, 1999.

[23] Emanuel F Petricoin, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, and Lance A Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359:572–577, 2002.

[24] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[25] Michele Sebag. Delaying the choice of bias: A disjunctive version space approach. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 444–452. Morgan Kaufmann, 1996.

[26] Dinesh Singh, Phillip G. Febbo1, Kenneth Ross, Donald G. Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A. Renshaw, Anthony V. D'Amico, Jerome P. Richie, Eric S. Lander, Massimo Loda, Philip W. Kantoff, Todd R. Golub, and William R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, March 2002.

[27] Evgueni N Smirnov and Peter J Braspenning. Version space learning with instance based boundary sets. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence*, pages 460–464, Brighton, UK, 1998.

[28] Lev A Soinov, Maria A Krestyaninova, and Alvis Brazma. Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biology*, 4:R6.1–R6.10, 2003.

[29] Nikunj V. Somia and Inder M. Verma. Gene therapy: Trials and tribulations. *Nature Reviews Genetics*, 1(2):91–99, 2000.

[30] Joachim Theihaber, Timothy Connolly, Sergio Roman-Roman, Steven Bushnell, Amanda Jackson, Kathy Call, Teresa Garcia, and Roland Baron. Finding genes in the c2c12 osteogenic pathway by k-neirest-neighbor classification of expression data. *Genome Research*, 12:165–176, 2002.

[31] Clare E. Thomas, Anja Ehrhardt, and Mark A. Kay. *Nature Reviews Genetics*, 4:346 –358, 2003.

[32] Eng-Juh Yeoh, Mary E. Ross, Sheila A. Shurtleff, W. Kent Williams, Divyen Patel, Rami Mahfouz, Fred G. Behm, Susana C. Raimondi, Mary V. Relling, Anami Patel, Cheng Cheng, Dario Campana, Dawn Wilkins, Xiaodong Zhou, Jinyan Li, Huiqing Liu, Ching-Hon Pui, William E. Evans, Clayton Naeve, Limsoon Wong, and James R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1:133–143, 2002.

[33] X. Zhang, G. Dong, and K. Ramamohanarao. Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 310–314, New York, August 2000.