A High Growth-Rate Emerging Pattern for Data Classification in Microarray Databases

Ye-In Chang, Zih-Siang Chen, and Tsung-Bin Yang
Dept. of Computer Science and Engineering, National Sun Yat-Sen University
Kaohsiung, Taiwan, R.O.C.
Email: changyi@cse.nsysu.edu.tw

***Abstract*—In the data classification problem for microarray datasets, we consider two biology datasets which reflect two extreme different classes for the given same sets of tests. Basically, the classification process contains two phases: (1) the training phase, and (2) the testing phase. The propose of the training phase is to find the representative Emerging Patterns (EPs) in each of these two datasets, where an EP is an itemset which satisfies some conditions of the growth rate from one dataset to another dataset. Note that the growth rate represents the differences between these two datasets. The EJEP strategy considers only those itemsets whose growth rates are infinite, since it claims that those itemsets may result in the high accuracy. However, the EJEP strategy will not keep those useful EPs whose growth rates are very high but not infinite. But, the real-world data always contains noises. The NEP strategy considers noises and provides the higher accuracy than the EJEP strategy. However, it still may miss some itemsets with high growth rates, which may result in the low accuracy. Therefore, in this paper, we propose a High Growth-rate EP (HGEP) strategy to improve the accuracy of the NEP strategy. From the performance study, our HGEP strategy shows the higher accuracy than the NEP strategy.**

***Index Terms*—classification, data mining, emerging pattern, gene expression, microarray**

I. Introduction

After 10 years of research and an amazing 2 billion dollars in funds, the Human Genome Project finally reported that 99% of the human genome had been sequenced [1], [2]. Based on the huge gene expression databases from the biological experiments, scientists refer to the expression analysis called microarrays. A DNA microarray [3], [4] is a collection of microscopic DNA spots attached to a solid surface, which is a technology for simultaneously profiling the expression levels of thousands of genes in a patient sample. Gene expression datasets are typically organized as a matrix. Assume that such a matrix has n rows of genes and m columns of conditions, where n is usually in the range of [2000, 20000] and m is usually in the range of [10, 100], *i.e.*, $n>>m$.

For *Emerging Patterns* (EPs), we will build a classification model from the training data, where the model is represented by n sets of EPs, one set per class. The model can be used to prodict unknown instances in the future. We stress that the building of the model, which is equivalent to the discovery of EPs.

According to different types of the training data, the strategies of the EPs can be divided into two categories, *i.e.*, the EPs with the infinite growth rate and the EPs with the finite growth rate. The EJEP strategy [5] only cares about those itemsets with the infinite growth rate. It ignores those patterns which have very large growth rates, although not infinite, *i.e.*, the so called "noise". However, the real-world data always contains noises and the NEP strategy [6] considers noises and provides higher accuracy than the EJEP strategy. Although the NEP strategy takes noise patterns into consideration, it still will miss some itemsets with a large growth rate, which may result in the low accuracy. Therefore, in this paper, we propose a High Growth-rate EP (HGEP) strategy to improve the disadvantage of the NEP strategy. From the experiment results, we show that the average accuracy of our HGEP strategy is higher than that of the NEP strategy.

II. The Related Work

In this section, we describe three well-known strategies for mining all kinds of Emerging Patterns [5], [6], [7], [8].

In [7], [8], they proposed a border-based algorithm for generating Emerging Patterns. Borders are used to represent candidates and subsets of EPs. A border is a structure, consisting of two bounds. A simple example might be < { {a}, {b}, { {a, b, c }, { b, d } } >. It represents all those sets which are supersets of {a} or {b} and subsets of {a, b, c} or {b, d}. In fact, the entire process of discovering EPs only needs to deal with borders.

A Jumping Emerging Pattern (JEP) [9], [10] only concerns those itemsets whose growth ratios are infinite. However, even the most efficient algorithm for mining JEPs [9] are not fast enough yet. It is reported that for the UCI Waveform dataset, which consists of 5000 instances by 21 attributes, it took up to four hours to mine 4096477 JEPs [9]. Therefore, in [5], they proposed Essential Jumping Emerging Patterns (EJEPs) to capture the crucial difference between a pair of data classes. EJEPs are defined as minimal itemsets whose supports in one data class are zero, but in another are above a given support threshold.

Manuscript received Nov 20, 2012; revised Jan 27, 2013.

doi: 10.12720/lnit.1.1.6-10

EJEPs allow noise tolerance in dataset D2. However, real-world data always contains noises in both dataset D1 and dataset D2. Both JEPs and EJEPs cannot capture those useful patterns whose support in dataset D1 is very small but not strictly zero; that is, they appear only several times due to random noises. Therefore, in [6], the Noise-tolerant EPs (NEPs) was proposed. The relationships of those patterns are shown in Fig. 1.

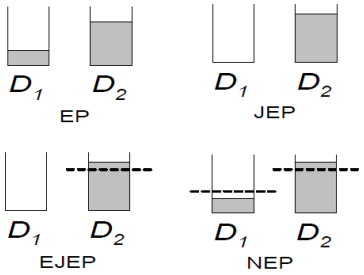

Figure 1. The illustration of all kinds of EPs.

III. The HGEP Classifier

A. The proposed strategy

To provide EPs with the high growth rate (*GR*), we take an itemset *X* which satisfies the following condition into consideration: $GR(propersubset(X)) < GR(X)$. If an itemset *X* satisfies the above condition, we keep the itemset which has longer length and a higher growth rate than those of its subsets. Take Table I as an example, where $SuppD1(X)$ and $SuppD2(X)$ represent the support value of itemset *X* in dataset *D*1 and dataset *D*2, respectively. Let X be {a, b} and its subset be {b}. Although itemset {b} has a higher support than itemset {a, b} in dataset D1, itemset {b} has a smaller growth rate from dataset D1 to dataset D2 than that of itemset {a, b}.

TABLE I. An example of two itemsets

X	suppD1(X)	suppD1(X)	GR(X)
{a, b}	1	2000	2000
{b}	4	2000	500

Based on the above observation, we define a new kind of Emerging Patterns, *High Growth-Rate Emerging Pattern* (HGEP), which can improve the accuracy of a classifier. An itemset *X* is an HGEP for dataset *D*2 from dataset *D*1 to dataset *D*2, if *X* satisfies one of the following two conditions: where $\delta 1$ and $\delta 2$ are the support thresholds of the dataset *D*1 and *D*2.

Condition 1:

(1-1) $0 < suppD1(X) \leqq \delta 1$ and $suppD2(X) \geqq \delta 2$, where $\delta 1 << \delta 2$.

(1-2) $GR(propersubset(X)) < GR(X)$.

Condition 2:

(2-1) $suppD1(X) = 0$ and $suppD2(X) \geqq \delta 2$.

(2-2) Any proper subset of X does not satisfy Condition (2-1).

In Condition 1, HGEPs keep those itemsets with the finite growth rate. On the other hand, in Condition 2, HGEPs keep those itemsets with infinite growth rates. Moreover, Condition 1-1 represents that the HGEPs should have a large enough growth rate, while Condition 1-2 represents that the growth rate of HGEPs should be as large as possible. Basically, Condition 2 has the same definition as an EJEP. However, our new-added Condition 1 provides the high noise-tolerance. Therefore, HGEPs contain not only the improved NEPs but also EJEPs.

Fig. 2-(a) shows the illustration. The dashed line means that the growth rate of the itemset is smaller than infinite; that is, the growth rate is finite. This line represents the property of these itemsets which satisfy Condition 1. The HGEP generated by Condition 1 is the itemset whose length is as long as possible and the growth rate is as high as possible. On the other hand, at point $(0, \infty)$, it represents the itemsets with the infinite growth rate. At this point, all the HGEPs are minimal itemsets; that is, it represents those itemsets which satisfy Condition 2.

For example, we use the training datasets shown in Fig. 2-(b) as the same datasets for the following comparison, where the growth rate threshold GR of EP is 2 and the support threshold δ of the EJEP strategy is 2. Thresholds $\delta 1$ and $\delta 2$ of the NEP and the HGEP strategies are 4 and 2000, respectively. From Fig. 2-(c), we see that the numbers of EPs, JEPs, and EJEPs are decreased because that the conditions are more and more strict. Moreover, the condition becomes loose from the EJEP strategy to the NEP strategy, since the number of the patterns from the EJEP strategy to the NEP strategy may be increased. The HGEP strategy has the similar situation as the NEP strategy. The number of HGEPs (and NEPs) may be more than that of EJEPs. But the number of HGEPs and NEPs are not necessarily equal. The relationships among these five kinds of Emerging Patterns are shown in Fig. 3. They have the following properties:

EP ⊇ JEP ⊇ EJEP.

NEP ⊇ EJEP and HGEP ⊇ EJEP.

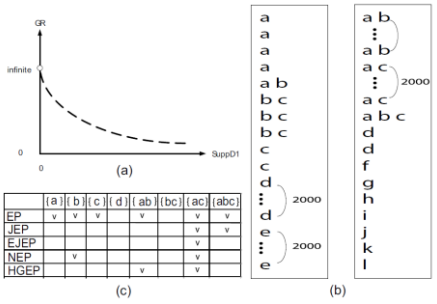

Figure 2. Illustration of the basic idea: (a) the illustration of HGEP; (b) an example of all kinds of EPs training datasets; (c) the result of five kinds of EPs.

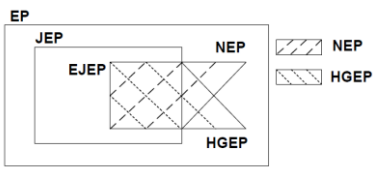

Figure 3. The relationships between various EPs.

B. The Contrast Pattern Tree Structure

In this subsection, we first define the order that we use to sort itemsets for adding them into the CP-tree. The order is important for building the tree structure and for traversing the tree systematically. It also helps to prune a huge search space in pattern discovery [6]. Next, we describe the data structure for mining HGEPs. Then, we present our algorithm for mining HGEPs.

For the Ordered List, we assume that the training datasets D contains dataset D1 and dataset D2. Let I = {i_1, i_2, …, i_n}be the set of all items appearing in the datasets D. Note that for an item $i \in I$, we have a singleton itemset $\{i\} \subset I$.

Let the minimum support threshold ξ be a positive real number. The support ratio of an item i between dataset D1 and dataset D2, denoted as SupportRatio(i) [6], is defined as *SupportRatio*(i) =

$$\begin{cases} 0: & \text{if } suppD1(\{i\}) < \xi \wedge suppD2(\{i\}) < \xi; \\ \infty: & \text{if } (suppD1(\{i\}) = 0 \wedge suppD2(\{i\}) \geq \xi) \\ & \text{or } (suppD1(\{i\}) \geq \xi \wedge suppD2(\{i\}) = 0); \\ max(\frac{suppD2(\{i\})}{suppD1(\{i\})}, \frac{suppD1(\{i\})}{suppD2(\{i\})}): & \text{otherwise.} \end{cases}$$

The *SupportRatio*(i) is used to capture individual items which represent a sharp contrast between dataset $D1$ and dataset $D2$ in the either direction. The larger the support ratio of an item is, the sharper the discriminating power associated with the item is. Usually, the support ratio is greater than or equal to 1, since we always permit the larger support to be divided by the smaller support. The support ratio will become 0, if both the supports in dataset $D1$ and dataset $D2$ are less than the minimum support threshold ξ. Items with a support ratio 0 are not useful for the HGEP mining, because HGEPs must satisfy the minimum support threshold ξ and HGEPs will never contain items whose supports in dataset $D1$ and dataset $D2$ are less than ξ. Note that for an item $i \in I$, if *SupportRatio*(i) = ∞, the item $\{i\}$ is an EJEP.

Based on the above definition, we can sort the itemsets by the total order $\prec$. Let i and j be two items. We say that $i \prec j$, if *SupportRatio*(i) > *SupportRatio*(j); or if *SupportRatio*(i) = *SupportRatio*(j) and $i < j$ (in the lexicographical order). Intuitively, [a_1, a_2, …, a_m] $\prec$ [b_1, b_2, …, b_n] means that the items in the former itemset have higher support ratios than those in the latter ones.

A *Contrast Pattern tree* (CP-tree) is an ordered multiway tree structure. Each node X of the CP-tree has a variable number of items, denoted as $X.items[i]$, where i = 1, 2, …, $X.itemNumber$, and $X.itemNumber$ is the number of items at node X [5]. If $X.itemNumber = k$ ($k > 0$), X has k itemsets from dataset D1, k itemsets from dataset D2, and at most k branches (child nodes), denoted as $X.counts$D1[i], $X.counts$D2, and $X.childe[i]$, respectively, where i = 1, 2, …, k. For $X.items[i]$ ($1 \leqq i \leqq k$), $X.counts$D1[i] records the number of itemsets in dataset D1 represented by the part of the path reaching $X.item[i]$, $X.counts$D2[i] records the number of itemsets in dataset D2 represented by the part of the path reaching $X.item[i]$, and $X.childs[i]$ refers to the subtree with the parent of $X.items[i]$ (also called $X.items[i]$'s subtree). To keep the branches of X ordered, we require that the k items inside node X satisfy: $X.items[1] \prec X.items[2] \prec \ldots \prec X.items[k]$, where $\prec$ is the support-ratio-descending order defined above. To simplify the following discussion, we apply conventional concepts of trees [6].

C. The Mining HGEPs Process

In this subsection, we describe how to use the CP-tree to mine HGEPs [6]. We search the CP-tree by the depthfirst order. At the same time, we need to reconstruct the CP-tree by merging its internal structure to ensure that we can discover all the HGEPs.

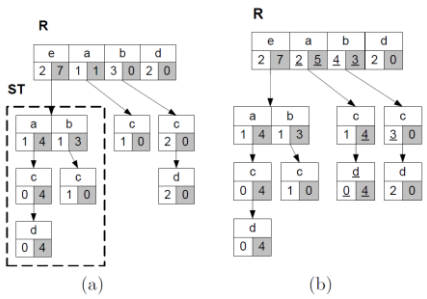

Figure 4. The CP-tree after merging nodes (*merge_tree*(*ST*, *R*)): (a) the original CP-tree; (b) the CP-tree after *merge_tree*(*ST*, *R*) and the underline parts denote the changes.

The merging process step merges the nodes of subtree T_1 into subtree T_2. Take Fig. 4 as an example. Let R be the root of the CP-tree and ST be the subtree of $R.e$. We perform a depth-first search of the CP-tree for HGEPs, which is equivalent to the exploration of the pattern space: {e}, {e, a}, {e, a, c}, {e, a, c, d} along path {e, a, c, d}; {e, b}, {e, b, c} along path {e, b, c}; {a}, {a, c} along path {a, c}; {b}, {b, c}, {b, c, d} along path {b, c, d} and {d} along path {d}. However, only the counts of {e}, {e, a}, {e, a, c}, {e, a, c, d} along path {e, a, c, d} are recorded obviously in the tree. Therefore, we need to merge subtree ST with the root R itself to adjust the counts of item a in the root R. We call procedure *merge_tree*(*ST*, *R*) shown in Figure 4 to do the merge operation. For the same reason, in the subtree of node $R.e$, the counts of b may not be correct. By merging nodes through the process of the depth-first search, we can make sure that the counts will be correctly calculated for determining HGEPs. Basically, the process merges all the nodes of ST into corresponding parts of R. In the process of mining HGEPs, training itemsets are sorted by its support ratios between both datasets. When inserting the ordered lists into the CP-tree, items with the high support-ratio, which are more likely to appear in an HGEP, are closer to the root. This makes the CP-tree compact, and decrease the requirement of the storage space of the tree. The mining process from a path in the CP-tree to an itemset is a one-to-one mapping. Using the predefined order $\prec$, we can generate the complete set of paths (itemsets) systematically through the depth-first search of the CP-tree.

According to the definition of HGEP, we divide procedure *mine_tree*() into two parts: mining HGEPs of Condition 1 and Condition 2. The flowchart of mining

HGEPs is shown in Fig. 5. In the part of Condition 1, all the HGEPs are the itemsets with the finite growth rate. For dataset D2, we can find the candidate of the form *Case* 1*a*. Similarly, for dataset D1, we can find the candidate of the form *Case* 1*b*. Moreover, these candidates are not necessary valid HGEPs. We have to compare the growth rate of each candidate with that of each of its subsets due to the definition of an HGEP.

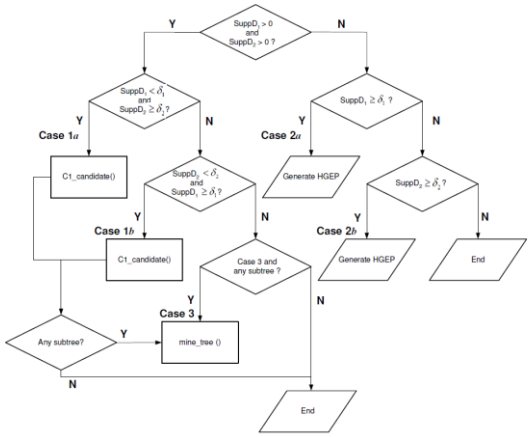

Figure 5. The flowchart of mining HGEPs in procedure mine_tree.

IV. Performance

In In this section, we study the performance of the HGEP strategy under the same input datasets: the UCI Machine Learning Repository:

(www.ics.uci.edu/~mlearn/MLRepository.html)

TABLE II. A COMPARISON OF ACCURACY

Case	Dataset	*NEP*	*HGEP*
1	australian	67.00	**80.99**
2	diabetes	63.33	**64.99**
3	glass2	**61.66**	**61.66**
4	heart	71.66	**86.66**
5	monk1-bin	78.33	**81.66**
6	monk3-local	71.66	**76.66**
7	mux6	48.33	**60.00**
8	parity5+5	**76.66**	73.33
9	pima	44.99	**61.66**
10	sonar	61.66	**66.66**
11	vehicle	**71.66**	**71.66**
12	waveform-21	**56.67**	**56.67**
13	waveform-40	**76.66**	68.33
	average	65.02	**70.07**

The comparison of the accuracy between our HGEP strategy and the NEP strategy is shown in Table II. We observe that our HGEP strategy ptovides the higher accuracy than the NEP strategy on 8 datasets (datasets No. 1, 2, 4, 5, 6, 7, 9, and 10). For the remaining 5 datasets (datasets No. 3, 8, 11, 12, and 13), the accuracy of the HGEP strategy is still very close to that of the NEP strategy. Therefore, the average accuracy of our HGEP strategy is still better than that of the NEP strategy. In the following cases, we add random noises to three datasets and observe how the accuracy is affected by the percentage of the increasing noises between our HGEP strategy and the NEP strategy. The percentages of noises of each dataset are chosen from 0% to 26%, which adds the number of noises from 1 instance to 9 instances into original 25 instances, respectively. Therefore, both of the NEP strategy and our strategy will not run out of memory.

In Fig. 6 and Fig. 7, we show the comparison, when we use *diabetes* (Case 2) and *mux6* (Case 7) as the input datasets, respectivedly. From both figures, we show that the accuracy of our HGEP strategy is better than that of the NEP strategy.

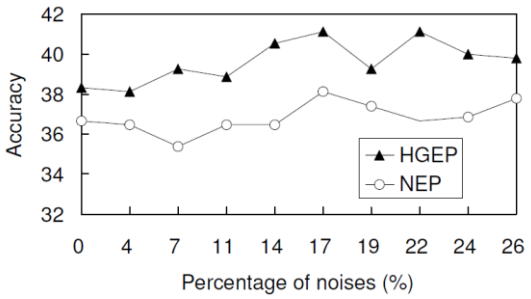

Figure 6. The effect of increasing noises on dataset *diabetes* (Case 2).

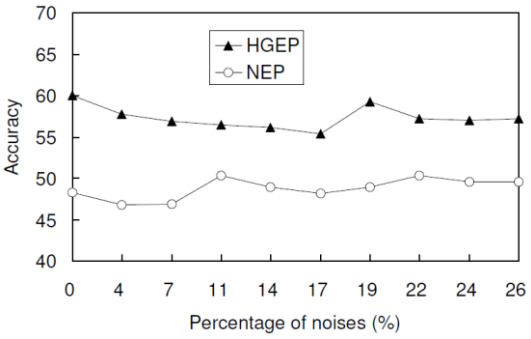

Figure 7. The effect of increasing noises on dataset *mux6* (Case 7).

V. Conclusion

Mining the significant differences based on the classifier from two-polarization classes of the gene expression recorded in microarray is an important task, such as the Emerging Pattern. In this paper, we have proposed a new strategy EP, called the HGEP strategy, for mining high growth-rate EPs. Based on the comparison with the NEP strategy by using several real microarray datasets, we have shown that the accuracy of our HGEP strategy is higher than that of the NEP strategy.

Acknowledgment

This research was supported in part by the National Science Council of Republic of China under Grant No. NSC-99-2221-E-110-080-MY3.

REFERENCES

[1] C. Gonzaga-Jauregui, J. R. Lupski, and R. A. Gibbs, "Human Genome Sequencing in Health and Disease," *Annual Review of Medicine*, vol. 63, pp. 35-61, February 2012.

[2] M. N. Wass, A. David, and M. J. Sternberg, "Challenges for the Prediction of Macromolecular Interactions," *Current Opinion in Structural Biology*, vol. 21, no. 3, pp. 382-390, June 2011.

[3] H. Hatakeyama , E. Ito , M. Yamamoto, H. Akita, Y. Hayashi, K. Kajimoto, N. Kaji, Y. Baba, and H. Harashima, "A DNA microarray-based analysis of the host response to a nonviral gene carrier: A strategy for improving the immune response," *Molecular Therapy : The Journal of the American Society of Gene Therapy*, vol. 19, no. 8, pp. 1487-1498, August 2011.

[4] C. C. Li, H. Y. Lo, C. Y. Hsiang, and T. Y. Ho, "DNA microarray analysis as a tool to investigate the therapeutic mechanisms and drug development of Chinese medicinal herbs," *BioMedicine*, vol. 2, no. 1, pp. 10-16, March 2012.

[5] H. Fan and K. Ramamohanarao, "An Efficient Single-Scan Algorithm for Mining Essential Jumping Emerging Patterns for Classification," in *Proc. 6th Pacific- Asia Conf. on Knowledge Discovery and Data Mining*, Taipei, 2002, pp. 456-462.

[6] H. Alhammady and K. Ramamohanarao, "Fast Discovery and the Generalization of Strong Jumping Emerging Patterns for Building Compact and Accurate Classifiers," *IEEE Trans. on Knowledge and Data Eng.*, vol. 18, no. 6, pp. 721-737, June 2006.

[7] G. Dong and J. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences," in *Proc. Int. Conf. Knowledge Discovery and Data Mining*, San Diego, 1999, pp. 43-52.

[8] G. Dong and J. Li, "Mining Border Descriptions of Emerging Patterns from Dataset Pairs," *Knowledge and Information Systems*, vol. 8, no. 2, pp. 178-202, August 2005.

[9] J. Bailey, T. Manoukian, and K. Ramamohanarao, "Fast Algorithms for Mining Emerging Patterns," in *Proc. 6th European Conf. on Principles and Practice of Knowledge Discovery in Databases*, Helsinki, 2002, pp. 39-50.

[10] J. Li, G. Dong, and K. Ramamohanarao, "Making use of the most expressive jumping emerging patterns for classification," *Knowledge and Information Systems*, vol. 3, no. 2, pp. 131-145, May 2001.

Ye-In Chang received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1986, and M.S. and Ph.D. degrees in computer science and engineering from The Ohio State University, Columbus, Ohio, in 1987 and 1991, respectively. From August 1991 to July 1999, she jointed the faculty of Department of Applied Mathematics at National Sun Yat-Sen University, Kaohsiung, Taiwan. Since August 1997, she has been a Professor in Department of Applied Mathematics at National Sun Yat-Sen University. Since August 1999, she has been a Professor in Department of Computer Science and Engineering at National Sun Yat-Sen University. Her research interests include database systems, distributed systems, data mining and bioinformatics.

Zih-Siang Chen received B.S. and M.S. degrees in computer science from National Pingtung University of Education in 2008 and 2010, respectively. He is currently a Ph.D. student in Department of Computer Science and Engineering at National Sun Yat-Sen University. His research interests include spatial mining and bioinformatics.

Tsung-Bin Yang received the B.S. degree in applied mathematics from National Chiayi University in 2005, and the M.S. degree in computer science and engineering from National Sun Yat-Sen University in 2007. He is currently a system design engineer in Taiwan.