Itemset Classified Clustering

Jun Sese[1] and Shinichi Morishita[2]

[1] Undergraduate Program for Bioinformatics and Systems Biology,
Graduate School of Information Science and Technology, University of Tokyo
sesejun@cb.k.u-tokyo.ac.jp

[2] Department of Computational Biology, Graduate School of Frontier Sciences,
University of Tokyo and Institute for Bioinformatics and Research and Development,
Japan Science and Technology Corporation
moris@cb.k.u-tokyo.ac.jp

Abstract. Clustering results could be comprehensible and usable if individual groups are associated with characteristic descriptions. However, characterization of clusters followed by clustering may not always produce clusters associated with special features, because the first clustering process and the second classification step are done independently, demanding an elegant way that combines clustering and classification and executes both simultaneously.

In this paper, we focus on itemsets as the feature for characterizing groups, and present a technique called "itemset classified clustering," which divides data into groups given the restriction that only divisions expressed using a common itemset are allowed and computes the optimal itemset maximizing the interclass variance between the groups. Although this optimization problem is generally intractable, we develop techniques that effectively prune the search space and efficiently compute optimal solutions in practice. We remark that itemset classified clusters are likely to be overlooked by traditional clustering algorithms such as two-clustering or k-means, and demonstrate the scalability of our algorithm with respect to the amount of data by the application of our method to real biological datasets.

1 Introduction

Progress in technology has led to the generation of massive amounts of data, increasing the need to extract informative summaries of the data. This demand has led to the development of data mining algorithms, such as clustering [14, 22, 9, 1, 5, 20], classification [4, 15], and association rules [2, 10, 13]. Recent technological progress in biology, medicine and e-commerce marketing has generated novel datasets that often consist of tuples represented by features and an objective numeric vector. For understanding what causes individual groups of data similar in terms of vectors, it is helpful to associate features with each group. A typical example from molecular biology is association of gene-controlling mechanisms with genes having analogous expression patterns. Such novel data motivate us to develop itemset classified clustering.

J.-F. Boulicaut et al. (Eds.): PKDD 2004, LNAI 3202, pp. 398–409, 2004.

1.1 Motivating Example

We here present a motivating example for showing the difference between traditional approach and our method called "itemset classified clustering."

Consider eight tuples $t_1, ..., t_8$ in Table 1. Each tuple contains feature items $i_1, ..., i_5$ and objective attributes a_1 and a_2. Fig. 1(A) shows objective vectors (a_1, a_2) of the tuples represented by white circles. Each tuple is at regular interval on the same square. For example, tuple t_2 locates at $(1, 2)$.

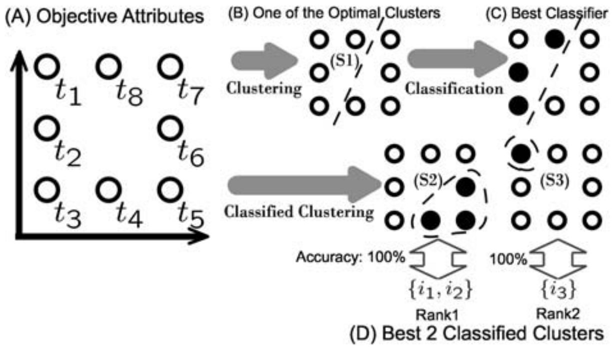


Fig. 1. Motivating Example of Itemset Classified Clustering

Table 1. Example Table

	Feature Items					Objective Attributes	
	i_1	i_2	i_3	i_4	i_5	a_1	a_2
t_1	0	0	1	0	0	1	3
t_2	1	0	0	1	1	1	2
t_3	0	0	0	1	1	1	1
t_4	1	1	0	0	1	2	1
t_5	1	1	0	0	1	3	1
t_6	1	1	0	0	1	3	2
t_7	0	0	0	1	0	3	3
t_8	0	1	0	1	1	2	3

Let us form clusters by dividing the tuples into two groups, S and $\bar{S}$, so that the division optimizes a proper measure. As the measure, one may utilize various values such as the diameter or the connectivity of a group, we here use interclass variance extended to the multi-dimension, which is common measure grounding in statistics to evaluate clusters. For simplicity, we call this multi-dimensional version just *interclass variance* in this paper. Let $c(S)$ denote the centroid of objective vector of S; namely, $\sum_{\boldsymbol{x} \in S} \boldsymbol{x}/|S|$. Interclass variance is defined as: $|S| \left|c(S) - c(S \cup \bar{S})\right|^2 + |\bar{S}| \left|c(\bar{S}) - c(S \cup \bar{S})\right|^2$. One of the solutions that maximize interclass variance is indicated by dotted straight line (S1) in Fig. 1(B). Line (S1) divides the tuples into cluster $S = \{t_1, t_2, t_3, t_8\}$ and cluster $\bar{S} = \{t_4, t_5, t_6, t_7\}$.

To understand the reason why tuples in each cluster are close, traditional clustering-classification approach such as conceptual clustering [12] attempts to find classifiers that are able to exactly classify the clusters using, say, additional feature items $i_1, ..., i_5$. In Table 1, "1" denotes the presence of an item in each tuple, while "0" denotes the absence. For instance, tuple t_2 includes item i_1, i_4 and i_5. In Fig. 1(C), solid black circles indicate tuples that contain itemset

$\{i_4, i_5\}$. Note that three out of four circles on the left of (S1) include the itemset, while none of the four on the right does. From this observation, one may derive the classifier that circles contain $\{i_4, i_5\}$ if and only if they are on the left of (S1), which holds seven out of eight cases, namely 87.5% accuracy. However, use of optimal clustering, such as the division by (S1), may not be able to identify a clustering so informative that each cluster is associated with its special feature items.

For identification of such beneficial clusters, our itemset classified clustering computes optimal clusters under the restriction that allows only splits expressible by a common itemset. We call such clusters *classified clusters*. (S2) in Fig. 1(D) indicates an example of classified cluster because the group $\{t_4, t_5, t_6\}$ is equal to the set of tuples that contain both i_1 and i_2. In other words, classifier $\{i_1, i_2\}$ has 100% accuracy for cluster $\{t_4, t_5, t_6\}$. $\{t_1\}$ is another example of classified cluster because the cluster is associated with the special classifier $\{i_3\}$ ((S3) in Fig. 1(D)). In these classified clusters, the set of tuples whose interclass variance is larger would be better cluster. For example, $\{t_4, t_5, t_6\}$ split by (S2) is better classified cluster than $\{t_1\}$ split by (S3). Note that the classified clusters are overlooked by two-clustering in Fig. 1(B), and the groups would not be found by general clustering algorithms such as k-means clustering.

One may wonder that the itemset associated with any optimal classified cluster is a closed pattern [21], which is a maximal itemset shared in common by transactions including the itemset. However, this claim is not true. For instance, $\{i_1, i_2\}$, which is not a closed pattern, classifies optimal classified cluster $\{t_4, t_5, t_6\}$. Its superset $\{i_1, i_2, i_5\}$ is a closed pattern and it also classifies the same optimal cluster; however, its inclusion of i_5 is superfluous, because neither of its subsets having i_5, namely $\{i_1, i_5\}$ and $\{i_2, i_5\}$, identifies any optimal classified cluster. This observation indicates that $\{i_1, i_2\}$ better classifies the optimal cluster. On the other hand, the closed pattern $\{i_4, i_5\}$ corresponds to non-optimal cluster $\{t_2, t_3, t_8\}$. Consequently, itemsets for optimal classified clusters and closed pattern itemsets are orthogonal notions.

This example reveals us that it is a non-trivial question to compute the optimal classified clusters because of two major problems. First, cluster that maximizes the index such as interclass variance is not always associated with special features. In our example, although cluster segmented by (S1) has the optimal index, the clusters are not associated with special features. Thus, the approach of clustering followed by classification is not effective for deriving classified clusters. Second, the number of combinations of items explodes as the number of items increases.

1.2 Related Work

On clustering-classification approach, refinement of clustering or classification might improve accuracy. Clustering studies have paid a great deal of attention to the choice of a measure that is tailored to the specificity of given data. For example, measures of sub-clustering for gene expression profiles [5, 20] and model-based measures [19] have been proposed. However, in these approaches, feature items are not supposed to be used to output directly constrained clusters.

Improvement of classification would increase the accuracy of the classifier for each cluster. This sophistication, however, could increase the description length of the classifier. For example, the number of nodes in a decision tree such as CART [4] and C4.5 [15] is likely to huge, making it difficult to understand. Moreover, classification methods do not generate classified clusters of similar objects in terms of numeric vectors associated with tuples.

2 Itemset Classified Clustering

In this section, we formalize the itemset classified clustering problem.

Itemset Classified Clustering: Suppose that we classify a tuple by checking to see whether it includes a feature itemset (e.g., $\{i_1, i_2\}$). Compute the optimal classifier that maximizes interclass variance with its corresponding cluster, or list the most significant N solutions.

In the running example, the optimal classifier is the itemset $\{i_1, i_2\}$ and its corresponding cluster is $\{t_4, t_5, t_6\}$. Furthermore, when $N = 10$, the itemset classified clustering problem demands the extraction of ten optimally classified clusters.

Unfortunately, it is difficult to compute an optimal itemset that maximizes the interclass variance, because the problem is NP-hard if we treat the maximum number of items in an itemset as a variable. The NP-hardness can be proved by reduction of the difficulty of the problem to the NP-hardness of finding the minimum cover [8]. The reduction consists of the following three major steps: (1) Treat a tuples as a vertex, and an item as a hyperedge enclosing such vertexes (tuples) that contain the item. (2) A cover is then expressed as an itemset. (3) An optimal itemset is proved to coincide with a minimal cover of vertexes according to the convexity of the interclass variance function [13].

The NP-hardness prompts us to make an effective method to compute the optimal itemset in practice. To compute the itemset classified clustering problem, we present the properties of interclass variance in the next section.

3 Interclass Variance

3.1 Basic Definitions

In this section, we first introduce the index, interclass variance.

Definition 1 Let D be the set of all tuples. Let i_k denote an item. We treat m numerical attributes in the given database as special, and we call these attributes *objective attributes.* Let $a_1, a_2, \ldots, a_m$ denote the objective attributes. Let $t[a_i]$ indicate the value of an objective attribute a_i associated with a tuple t. ■

In Table 1, let $D = \{t_1, t_2, \ldots, t_8\}$ and $i_1, \ldots, i_5$ be items, and a_1 and a_2 be objective attributes. Then, t_2 contains itemset $\{i_1, i_4, i_5\}$ and $t_2[a_1] = 1$ and $t_2[a_2] = 2$.

We divide D into two groups using itemset I, D_I and $\bar{D_I}$. D_I means a set of tuples that include itemset I, and $\bar{D_I}$ is the complement of D_I; namely $D - D_I$. In the running example, when $I = \{i_1\}$, $D_I = \{t_2, t_4, t_5, t_6\}$ and $\bar{D_I} = \{t_1, t_3, t_7, t_8\}$.

Definition 2 Let n be $|D|$ and $x(I)$ be $|D_I|$. Let s_i be $\sum_{t \in D} t[a_i]$, and $y_i(I)$ be $\sum_{t \in D_I} t[a_i]$. We define the interclass variance of itemset I as

$$x(I) \sum_{i=1}^{m} \left(\frac{y_i(I)}{x(I)} - \frac{s_i}{n} \right)^2 + (n - x(I)) \sum_{i=1}^{m} \left(\frac{s_i - y_i(I)}{n - x(I)} - \frac{s_i}{n} \right)^2 . \blacksquare$$

Since s_i and n are independent of the choice of itemset I according to the definition of interclass variance, the values of $x(I)$ and $y_i(I)$ uniquely determine interclass variance. Therefore, we will refer to interclass variance as $var(x, y_1, \ldots, y_m)$.

Definition 3 $var(x, y_1, \ldots, y_m) = x \sum_{i=1}^{m} \left(\frac{y_i}{x} - \frac{s_i}{n} \right)^2 + (n - x) \sum_{i=1}^{m} \left(\frac{s_i - y_i}{n - x} - \frac{s_i}{n} \right)^2 . \blacksquare$

In the running example, let $I = \{i_1\}$. $n = 8$, $s_1 = s_2 = 16$, $x(I) = 4$, $y_1(I) = 9$ and $y_2(I) = 6$. Therefore, $var(x(I), y_1(I), y_2(I)) = 2.5$.

When $m = 1$, this measure equals the interclass variance, a well-known statistical measure. Therefore, this index is a multi-dimensional generalization of the interclass variance.

From the definition of interclass variance, we can prove the convexity of $var(x, y_1, \ldots, y_m)$. The convexity is useful for conducting an effective search for significant itemsets.

Definition 4 A function $f(x, y_1, \ldots, y_m)$ is convex if for any $(x, y_1, \ldots, y_m)$ and $(x', y'_1, \ldots, y'_m)$ in the domain of f, and for any $0 \leq \lambda \leq 1$,
$\lambda f(x, y_1, \ldots, y_m) + (1 - \lambda) f(x', y'_1, \ldots, y'_m) \geq f\left(\lambda(x, y_1, \ldots, y_m) + (1 - \lambda)(x', y'_1, \ldots, y'_m)\right) \blacksquare.$

Proposition 1 $var(x, y_1, \ldots, y_m)$ $(0 \leq x \leq n)$ is a convex function.

Proof *(Omitted)* $\blacksquare$

3.2 Upper Bound

To calculate the set of significant itemsets, it is useful to estimate an upper bound of the interclass variance of any superset J of I because the information allows us to restrict the search space of the itemsets. For example, if an upper bound of itemset $\{i_2\}$ is less than the interclass variance of $\{i_1\}$, then $\{i_2\}$ and its supersets (e.g., $\{i_2, i_3\}$) can be pruned.

To estimate an upper bound, first, we map each itemset $J \supseteq I$ to a tuple $(x(J), y_1(J), \ldots, y_m(J))$, which we call *stamp point* of J. Subsequently, we calculate a hyper-polyhedron that encloses all the stamp points of $J \subseteq I$ for the given itemset I. Finally, we prove that one of the vertexes on the wrapping hyper-polyhedron provides an upper bound. We now present precise definitions and propositions.

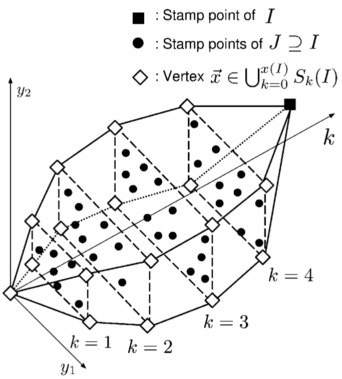

: Stamp point of I
: Stamp points of $J \supseteq I$
: Vertex $\vec{x} \in \bigcup_{k=0}^{x(I)} S_k(I)$
y_2
k
$k=4$
$k=3$
$k=1$
$k=2$
y_1

Fig. 2. The hyper-polyhedron surrounding all the stamp points of $J \supseteq I$

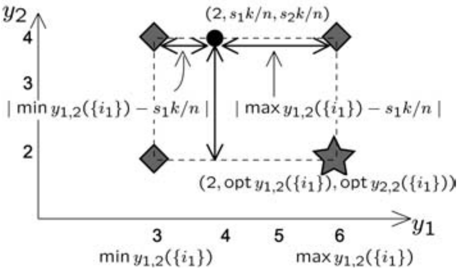

$y2$
4
3
2
$(2, s_1k/n, s_2k/n)$
$|\min y_{1,2}(\{i_1\}) - s_1k/n|$
$|\max y_{1,2}(\{i_1\}) - s_1k/n|$
$(2, \mathrm{opt}\, y_{1,2}(\{i_1\}), \mathrm{opt}\, y_{2,2}(\{i_1\}))$
3
4
5
6
$y1$
$\min y_{1,2}(\{i_1\})$
$\max y_{1,2}(\{i_1\})$

Fig. 3. The point maximizing interclass variance in $S_2(\{i_1\})$

Definition 5 Let $y_{i,k}(I)$ be the multi-set $\{\sum_{t\in D'\subseteq D_I} t[a_i] \mid |D'| = k\}$. Let $S_k(I)$ be $\{(k, z_1, \dots, z_m) \mid z_i = \max y_{i,k}(I) \text{ or } z_i = \min y_{i,k}(I) \text{ for } i = 1, 2, ..., m\}$, where m is the number of objective attributes. Each element in $S_k(I)$ is a vertex of the wrapping hyper-polyhedron on $x = k$. ■

Here, we describe $y_{i,k}(I)$ as a multi-set, because the multi-set representation will be required later to define the best N solutions.

For example, in Table 1, let I be $\{i_1\}$. Then, $D_I = \{t_2, t_4, t_5, t_6\}$, $y_{1,1}(I) = \{1, 2, 3, 3\}$, $y_{1,2}(I) = \{3, 4, 4, 5, 5, 6\}$. Furthermore, $y_{2,2}(I) = \{2, 3, 3, 3, 3, 4\}$. Therefore, $S_2(I) = \{(2, 3, 2), (2, 3, 4), (2, 6, 2), (2, 6, 4)\}$.

Lemma 1 For any itemset $J \supseteq I$,
$var(x(J), y_1(J), \dots, y_m(J)) \leq \max_{0\leq k\leq x(I)}\{var(\boldsymbol{x}) \mid \boldsymbol{x} \in S_k(I)\}$.

Proof It is known that any convex function is maximized at one of the vertexes on the boundary of a convex hyper-polyhedron [1]. From Proposition 1, interclass variance is a convex function. Due to its convexity, it is sufficient to prove that the hyper-polyhedron of $\bigcup_{k=0}^{x(I)} S_k(I)$ encloses all the stamp points of itemsets $J \supseteq I$.

Note that $D_J \subseteq D_I$ for any itemset $J \supseteq I$. Since $y_i(J) \in y_{i,x(J)}(I) (0 \leq i \leq m)$, $\min y_{i,x(J)}(I) \leq y_i(J) \leq \max y_{i,x(J)}(I)$. Therefore, $var(x(J), y_1(J), \dots, y_m(J)) \leq \max\{var(\boldsymbol{x}) \mid \boldsymbol{x} \in S_{x(J)}(I)\} \leq \max_{0\leq k\leq x(I)}\{var(\boldsymbol{x}) \mid \boldsymbol{x} \in S_k(I)\}$. ■

According to this lemma, we can estimate an upper bound of the interclass variance of itemset $J \supseteq I$. Fig. 2 illustrates the wrapping strategy. We can confirm that the hyper-polyhedron surrounds all stamp points. Indeed, according to this lemma, we can estimate an upper bound of the interclass variance of any itemset $J \supseteq I$.

However, two problems may appear. First, the wrapping hyper-polyhedron might not be sufficiently tight to form a convex hull of $J \supseteq I$. Second, it could be too costly to calculate an upper bound when m is large because the number of vertices could be $x(I) \times 2^m$ in the worst case.

For the first problem, we show that our wrapping function is tight enough to solve real data in Section 5. To overcome the second problem, we develop a technique that allows us to dramatically reduce the number of vertices to consider in the next subsection.

3.3 Efficient Calculation of the Upper-Bound

We first remark that the vertex farthest away from $s_i k/n$ among the vertices of the hyper-polyhedron on hyper-plain $x = k$ maximizes the interclass variance. From this property, we will devise an efficient algorithm for searching an upper bound.

Definition 6 We denote the vertex farthest away from $s_i k/n$ by $(k, \mathrm{opt}\, y_{1,k}(I), \mathrm{opt}\, y_{2,k}(I), \ldots)$, where

$$\mathrm{opt}\, y_{i,k}(I) = \begin{cases} \min y_{i,k}(I) & \text{if } \mid \min y_{i,k}(I) - s_i k/n \mid > \mid \max y_{i,k}(I) - s_i k/n \mid \\ \max y_{i,k}(I) & \text{otherwise} \end{cases} \blacksquare.$$

In the running example, $k = 2$, $n = 8$ and $s_1 = s_2 = 16$. Then, $s_1 k/n = s_2 k/n = 4$. Since $y_{1,2}(\{i_1\}) = \{3, 4, 4, 5, 5, 6\}$, $\min y_{1,2}(\{i_1\}) = 3$ and $\max y_{1,2}(\{i_1\}) = 6$. Therefore, $\mathrm{opt}\, y_{1,2}(\{i_1\}) = \max y_{1,2}(\{i_1\}) = 6$. Similarly, $\mathrm{opt}\, y_{2,2}(\{i_2\}) = \min y_{2,2}(\{i_1\}) = 2$. The arrows in Fig. 3 illustrate the selection of opt.

Lemma 2 $var(k, \mathrm{opt}\, y_{1,k}(I), \ldots, \mathrm{opt}\, y_{m,k}(I)) = \max\{var(\boldsymbol{x}) \mid \boldsymbol{x} \in S_k(I)\}$.

Proof Let $\boldsymbol{c} = (k, s_1 k/n, \ldots, s_m k/n)$. $var(x, y_1, \ldots, y_m) = x \sum_{i=1}^{m} \left(\frac{y_i}{x} - \frac{s_i}{n}\right)^2 + (n - x) \sum_{i=1}^{m} \left(\frac{s_i - y_i}{n-x} - \frac{s_i}{n}\right)^2 = \left(\frac{1}{x} + \frac{1}{n-x}\right) \sum_{i=1}^{m} \left(y_i - \frac{s_i}{n} x\right)^2$. From this equality, on hyper-plain $x = k$, if $|(k, y_1, \ldots, y_m) - \boldsymbol{c}| \geq |(k, y_1', \ldots, y_m') - \boldsymbol{c}|$, then $var(k, y_1, \ldots, y_m) \geq var(k, y_1', \ldots, y_m')$.

Now, since $\mathrm{opt}\, y_{i,k}(I)$ denotes the value farthest away from $s_i k/n$ among $y_{i,k}(I)$, the point farthest from $\boldsymbol{c}$ on $x = k$ is $(k, \mathrm{opt}\, y_{1,k}(I), \ldots, \mathrm{opt}\, y_{m,k}(I))$. Therefore, $var(k, \mathrm{opt}\, y_{1,k}(I), \ldots, \mathrm{opt}\, y_{m,k}(I)) = \max\{var(\boldsymbol{x}) \mid \boldsymbol{x} \in S_k(I)\}$. ■

In the running example, $(2, \mathrm{opt}\, y_{1,2}(\{i_1\}), \mathrm{opt}\, y_{2,2}(\{i_1\})) = (2, \max y_{1,2}(\{i_1\}), \min y_{2,2}(\{i_1\}))$, and its stamp point is indicated with star in Fig. 3.

Lemma 1 and 2 lead to the following theorem.

Theorem 1 For any itemset $J \supseteq I$,

$$\begin{aligned} & var(x(J), y_1(J), \ldots, y_m(J)) \\ & \leq \max_{0 \leq k \leq x(I)} var(k, \mathrm{opt}\, y_{1,k}(I), \mathrm{opt}\, y_{2,k}(I), \ldots, \mathrm{opt}\, y_{m,k}(I)). \blacksquare \end{aligned}$$

Definition 7

$$u(I) = \max_{0 \le k \le x(I)} var(k, \mathrm{opt}\, y_{1,k}(I), \mathrm{opt}\, y_{2,k}(I), \ldots, \mathrm{opt}\, y_{m,k}(I)).\blacksquare$$

$var(2, \mathrm{opt}\, y_{1,2}(\{i_1\}), \mathrm{opt}\, y_{2,2}(\{i_1\})) = var(2, \max y_{1,2}(\{i_1\}), \min y_{2,2}(\{i_1\})) = 5.33$. Similarly, when $k = 1, 3$ and 4, we can calculate the interclass variances as 2.29, 6.93 and 2.5, respectively. Therefore, $u(\{i_1\}) = 6.93$.

Let us consider the effective computation of $u(I)$. We can calculate $\min y_{i,k}(I)$ ($\max y_{i,k}(I)$, resp) for each i by scanning the sorted list of $y_{i,1}(I)$ once. Therefore, the following lemma can be proved, and its pseudo-code used to calculate $u(I)$ is shown in Fig. 4. In the pseudo-code, $y_i^k(I)$ is k-th smallest value in $y_{i,1}(I)$. This pseudo-code confirms the following lemma.

Lemma 3 Let I be an itemset. The time complexity for calculating $u(I)$ is $O(mn \log n)$. ■

4 Itemset Classified Clustering Algorithm

The estimation of an upper bound enables us to design an algorithm to solve the itemset classified clustering problem as a result of the following pruning observation.

```
Calculate-u(Itemset $I$)
// Preprocessing $O(mn \log n)$
Sort $y_{i,1}(I)$ for each $i \in [1, m]$;
$u = 0$; // $u$ stores upper bound value.
// Select $\mathrm{opt}\, y_{i,k}(I)$. $O(mn)$
for each $k \in [1, x(I)]$ do
  for each $i \in [1, m]$ do
    $\min y_{i,k}(I) := \min y_{i,k-1}(I) + y_i^k(I)$;
    $\max y_{i,k}(I) := \max y_{i,k-1}(I) + y_i^{x(I)-k+1}(I)$;
    // Select $\mathrm{opt}\, y_{i,k}(I)$ according to Def. 6
    if $| \min y_{i,k}(I) - s_i k/n |$
        $>| \max y_{i,k}(I) - s_i k/n |$ then
      $\mathrm{opt}\, y_{i,k}(I) := \min y_{i,k}(I)$;
    else $\mathrm{opt}\, y_{i,k}(I) := \max y_{i,k}(I)$;
    end
  end
  $v := var(k, \mathrm{opt}\,_{1,k}(I), \ldots, \mathrm{opt}\,_{m,k}(I))$;
  $u := v$ if $v > u$; // Update upper bound $u$
end
Return $u$;
```

Fig. 4. The pseudo-code used to calculate $u(I)$

```
Itemset Classified Clustering
$(\mathcal{Q}_1, \mathcal{L})$ =ICC-init;
// $\mathcal{L}$: list of the best $N$ rules
$\mathcal{B}_1 := \mathcal{Q}_1$; $k := 1$;
repeat until $\mathcal{Q}_k = \phi$
  for each $B \in \mathcal{B}_1$, $Q \in \mathcal{Q}_k$
    st. tail($Q$) < head($B$) and $u(Q) \ge \tau(\mathcal{L})$
    // Search only productive $Q$
    // $\tau(\mathcal{L})$ : $N$th best value in $\mathcal{L}$
    if $u(B) < \tau(\mathcal{L})$
      // Disposal of unproductive 1-itemsets
      Remove $B$ from $\mathcal{B}_1$; next;
    end
    // Construct new candidate itemset
    // and update $\mathcal{Q}_{k+1}$ and $\mathcal{L}$
    $(\mathcal{Q}_{k+1}, \mathcal{L})$ := ICC-update($Q \cup B, \mathcal{Q}_{k+1}, \mathcal{L}$);
  end
  $k++$;
end
Return $\mathcal{L}$; // the best $N$ rules
```

Fig. 5. The pseudo-code for Itemset Classified Clustering

```
ICC-init
$\mathcal{L} := \phi$;
for each $I \in \{J | J$ is a 1-itemset $\}$
  // Calculate upper-bound
  // and $var$ for each 1-itemset
  $(\mathcal{Q}_1, \mathcal{L})$ := ICC-update($I, \mathcal{Q}_1, \mathcal{L}$);
end
Return $\mathcal{Q}_1$ and $\mathcal{L}$;
```

Fig. 6. The pseudo-code for ICC-init

```
ICC-update
(Itemset $I$, Set of Itemsets $\mathcal{Q}$, List of the best $N$ rules $\mathcal{L}$)
$u(I)$ =Calculate-u($I$); // Calculate $u(I)$
// Update $\mathcal{Q}$ and $\mathcal{L}$ if necessary (Observation 1)
if $u(I) \ge \tau(\mathcal{L})$
  Put $I$ into $\mathcal{Q}$;
  if $var(x(I), y_1(I), \ldots, y_m(I)) \ge \tau(\mathcal{L})$
    // Update the best list $\mathcal{L}$
    $\mathcal{L}$ := list of the best $N$ rules in $\mathcal{L} \cup \{I\}$;
  end
end
Return $\mathcal{Q}$ and $\mathcal{L}$;
```

Fig. 7. The pseudo-code for ICC-update

Table 2. Default parameters

Parameter	Meaning	Default Value
$\|D\|$:	The number of tuples (genes)	4,000
N:	The number of the best classifiers (clusters)	10
L:	The length of the promoter region	300

Table 3. Yeast Gene Expression Profile Dataset

Dataset	# of feature items	# of objective attributes	# of available genes (tuples)
Spellman [8]	86,016	23	4,347
Cho [6]	86,016	17	6,137
DeRisi [7]	86,016	7	5,882

Observation 1 [13] Let us evaluate itemsets using an index satisfying convexity. Let N be the user-specified number of itemset classified clustering rules. Let $\mathcal{L}$ be a list of the best N itemsets, and $\tau(\mathcal{L})$ be the N-th best value in $\mathcal{L}$. For any itemset $J \supseteq I$, since $u(J) \leq u(I)$, J can be pruned when $u(I) < \tau(\mathcal{L})$. ■

This observation enabled us to design the algorithm "itemset classified clustering." To describe the itemset classified clustering, we define the following notation.

Definition 8 Let k-itemset be an itemset containing k items. Let $\mathcal{Q}_k$ and $\mathcal{B}_1$ be a set of k-itemsets and a set of 1-itemsets, respectively. Let us assume that there exists a total order among the items. Let I be an itemset, and head(I)(tail(I), respectively) denote the minimum (maximum) number of items in I. ■

For example, $\{i_1\}$ is 1-itemset and $\{i_1, i_3\}$ is 2-itemset. Assuming that $i_1 \prec i_2 \prec \cdots$ and $I = \{i_2, i_3, i_4\}$. head$(I) = i_2$ and tail$(I) = i_4$.

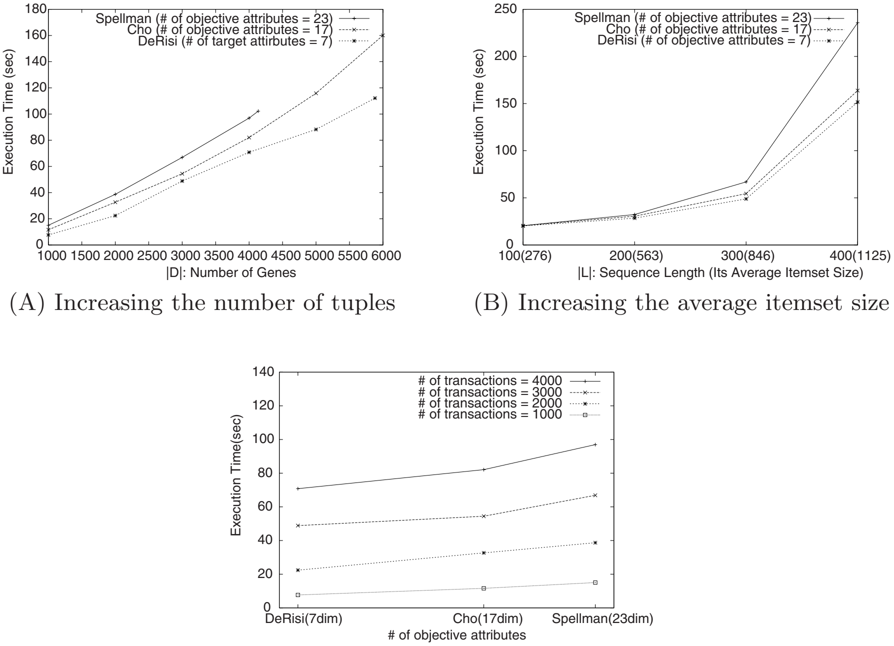
Fig. 5-7 shows the pseudo-code used in itemset classified clustering. In this pseudo-code, instead of traversing the itemsets over a lattice structure like apriori algorithm [2], we traverse them over a tree structure based on the set enumeration tree [3, 17], which is tailored to computing the best N rules using a statistical measure.

5 Experimental Results

5.1 Dataset

This section presents experimental results examining the effectiveness and performance of itemset classified clustering using yeast gene expression dataset and its DNA sequences. Gene expression dataset includes expression levels of thousands of yeast genes using DNA microarray under distinct conditions and range from 7 to 23 [18, 6, 7]. We consider each level of expression as an objective value, and each gene as a tuple.

Table 2 shows the parameters and their default values used to construct the test data. Table 3 summarizes the three microarray experiments objective values.



(A) Increasing the number of tuples

(B) Increasing the average itemset size

(C) Increasing the number of objective attributes

Fig. 8. Scalability of the performance

The experiments have 7 [7], 17 [6], and 23 [18] objective attributes, respectively. In our experimental results, since gene expression levels are regulated by specific combinations of short subsequences in promoter region (neighbour region of each gene), we use the existence of all the subsequences whose lengths are between six to eight, for instance, "AATGGC" or "AGATCGCC", as feature items. Therefore, the number of items is $4^6 + 4^7 + 4^8 = 86,016$ because a DNA sequence consists of 4 letters, A, C, G, and T. Briefly, we test itemset classified clustering algorithm on a database containing 1,000-6,000 tuples, 7-23 objective attributes, and 86,016 items. As shown in motivating example, itemset classified clustering can extract clusters which are different from clustering-classification approach, we compute best 10 clusters containing more than or equal to 10% of all tuples without any threshold of interclass variance.

We evaluated the overall performance of itemset classified clustering implemented in C with an Ultra SPARC III 900 MHz processor and 1 GB of main memory on Solaris 8.

5.2 Scalability

Two distinct ways of increasing the size of the dataset were used to test the scalability of itemset classified clustering. We represent the scalability in Fig. 8.

Fig. 8(A) illustrates the execution time when we increase the number of tuples $|D|$ from 1,000 to 6,000. The figure shows that the execution time scales almost linearly with the number of tuples for every dataset. Therefore, this figure indicates that our algorithm is scalable for increasing the tuples.

Fig. 8(B) demonstrates the performance of itemset classified clustering when the average number of items in itemsets increases for adding noisy tuples to dataset. Such a dataset can be obtained by increasing promoter length L because the probability of whether each short subsequence appears in the promoter region increases. In Fig. 8(B), L ranges from 100 to 400, while $|D| = 3,000$. This figure shows that the execution time increases quadratically to the average itemset size. This graph shows two types of effectiveness of our itemset classified clustering algorithm. One is that, since our measure, interclass variance, inherits the characteristics of statistical measures, itemset classified clustering effectively neglects noisy tuples. The other is that our upper-bound pruning is effective albeit the search space of itemsets grows more than exponentially according to increase of the average size of itemsets. This dramatic effects of pruning could be observed even when the objective attribute has over twenty dimensions. Indeed, our wrapping hyper-polyhedron of the stamp tuples of all the supersets of an itemset is bigger than their tight convex hull. Nevertheless, these experiments prove that our wrapping upper bound estimation is sufficient for a real dataset.

We converted Fig. 8(A) into Fig. 8(C) to study the effect of the number of objective attributes (dimensions). The figure shows that the execution time also scales almost linearly with the number of objective attributes.

6 Concluding Remarks

This paper presented the demand of the consideration of new data sets consisting of tuples which is represented by a feature itemset and an objective vector. To analyze the data, because traditional clustering-classification method may not always produce clusters associated with a feature itemset, we introduced a new paradigm, itemset classified clustering, which is a clustering that allows only splits expressible by a common feature itemset, and computes the optimal itemset that maximizes the interclass variance of objective attributes, or list the most significant N solutions. This itemset classified clustering can extract clusters overlooked by two-clustering or k-means clustering.

Our experimental results show that the itemset classified clustering has the scalability of performance for tuple size, objective attribute size, and itemset size. Therefore, the method can solve the real molecular biological problem containing 6,000 tuples, more than 80,000 boolean feature items and 23 numerical objective attributes.

Solving the itemset classified clustering problem is applicable to various problems because this problem prompts us to reconsider the results of both clustering and classification analysis. One example is to find the association between patients' gene expressions and their pathological features. [16] Furthermore, the replacement of itemset with other features such as numerical or categorical features might expand the application of clustering and classification algorithms.

References

1. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. of ACM SIGMOD 1998*, pages 94–105, 1998.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of 20th VLDB*, pages 487–499, 1994.
3. R. Bayardo. Efficiently mining long patterns from databases. In *Proc. of ACM SIGMOD 1998*, pages 85–93, 1998.
4. L. Breiman, R. A. Olshen, J. H. Friedman, and C. J. Stone. *Classification and Regression Trees.* Brooks/Cole Publishing Company, 1984.
5. Y. Cheng and G. M. Church. Biclustering of expression data. In *Proc. of the Eighth Intl. Conf. on ISMB*, pages 93–103, 2000.
6. R. J. Cho, M. J. Campbell, et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.
7. J. L. DeRisi, V. R. lyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.
8. M. R. Garey and D. S. Johnson. *Computer and Intractability. A Guide to NP-Completeness.* W. H. Freeman, 1979.
9. S. Guha, R. Rastogi, and K. Shim. Cure: an efficient clustering algorithm for large databases. In *Proc. of ACM SIGMOD 1998*, pages 73–84, 1998.
10. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. of ACM SIGMOD 2000*, pages 1–12, 2000.
11. R. Horst and H.Tuy. *Global optimization: Deterministic approaches.* Springer-Verlag, 1993.
12. R. S. Michalski and R. E. Stepp. *Learning from observation: Conceptual clustering*, pages 331–363. Tioga Publishing Company, 1983.
13. S. Morishita and J. Sese. Traversing itemset lattice with statistical metric pruning. In *Proc. of ACM PODS 2000*, pages 226–236, 2000.
14. R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *20th VLDB*, pages 144–155, Los Altos, CA 94022, USA, 1994.
15. J. R. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., 1993.
16. J. Sese, Y. Kurokawa, M. Monden, K. Kato, and S. Morishita. Constrained clusters of gene expression profiles with pathological features. *Bioinformatics*, 2004. in press.
17. J. Sese and S. Morishita. Answering the most correlated n association rules efficiently. In *Proc. of PKDD'02*, pages 410–422, 2002.
18. P. T. Spellman and other. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.
19. J. Tantrum, A. Murua, and W. Stuetzle. Hierarchical model-based clustering of large datasets through fractionation and refractionation. In *Proc. of the KDD '02*, 2002.
20. H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *Proc. of ACM SIGMOD 2002*, pages 394–405, 2002.
21. M. Zaki and C. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *2nd SIAM International Conference on Data Mining*, 2002.
22. T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. In *Proc. of ACM SIGMOD 1996*, pages 103–114, 1996.