In the Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems (ISMIS 2006), Lecture Notes in Artificial Intelligence (LNAI), volume 4203, pages 47-57, Bari, Italy, September 2006. Springer-Verlag.

Visualizing Transactional Data with Multiple Clusterings for Knowledge Discovery

Nicolas Durand[1] and Bruno Crémilleux[1] and Einoshin Suzuki[2]

[1] GREYC CNRS UMR 6072,
University of Caen Basse-Normandie, France
[2] Department of Informatics, ISEE,
Kyushu University, Fukuoka, Japan

Abstract. Information visualization is gaining importance in data mining and transactional data has long been an important target for data miners. We propose a novel approach for visualizing transactional data using multiple clustering results for knowledge discovery. This scheme necessitates us to relate different clustering results in a comprehensive manner. Thus we have invented a method for attributing colors to clusters of different clustering results based on minimal transversals. The effectiveness of our method VisuMClust has been confirmed with experiments using artificial and real-world data sets.

1 Introduction

Visualization is of great importance in data mining as the process of knowledge discovery largely depends on the user, who, as a human being, is said to obtain 80% of information from eyesight [7, 11, 19]. A lot of data mining methods address transactional data. Such data consist of a set of transactions each of which is represented as a set of items and are ubiquitous in real world especially in commerce. As the proliferation of association rule research shows, transactional data have long been an important target of data mining [2]. Attempts for learning useful knowledge from transactional data are numerous and include probabilistic modeling [6], association rule discovery [15], and itemset approximation [1], to name but a few.

Cadez et al. investigate application of probabilistic clustering to obtain profiles from transactional data [6]. They point out visualization as a promising application of their method and have extended their approach for visualization of web navigation patterns [5]. An independent attempt for visualizing time-series data with probabilistic clustering exists [18].

There are a large number of clustering algorithms in machine learning and this situation comes from the fact that the goodness of a clustering result in general depends on the subjective view of the user [10]. We hence believe that visualization based on multiple clustering results can possibly provide various views and outperform the visualization methods based on a single clustering result. In this manuscript, we tackle this issue for transactional data by resolving several difficulties including attribution of colors to clusters from multiple

clustering results. Our method differs from the meta-clustering and clustering aggregation approaches [12] because the color attribution satisfies specific properties for visualization.

The rest of this manuscript is structured as follows. Section 2 defines the problem of visualizing transactional data and provides a survey of related work. We propose our method VisuMClust in Section 3, demonstrate its effectiveness by experiments in Section 4, and conclude in Section 5.

2 Visualizing Transactional Data with Multiple Clustering Results

2.1 Definition of the Problem

Visualizing transactional data with multiple clustering results can possibly realize multiple-view analysis of the user. Such analysis is expected to be effective for a wide range of users. Here we formalize this problem for knowledge discovery.

Transactional data $\mathcal{T}$ consist of n transactions $t_1, t_2, \ldots, t_n$ i.e. $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ each of which is described as a set of items $\mathcal{I}$. A transaction t is a subset of $\mathcal{I}$ i.e. $t \subset \mathcal{I}$. A clustering algorithm i for transactional data outputs a set $\mathcal{C}_i$ of clusters $c_{i,1}, c_{i,2}, \ldots, c_{i,m}$ given $\mathcal{T}$ i.e. $\mathcal{C}_i = \{c_{i,1}, c_{i,2}, \ldots, c_{i,m}\}$. We also call $\mathcal{C}_i$ a clustering result of the clustering algorithm i. A clustering algorithm where each transaction belongs to only one cluster is called a *crisp* clustering algorithm, otherwise a *soft* clustering (an overlapping between clusters may appear).

The display result D of an information visualization method is diverse and here we avoid restricting its possibilities. We define that our multi-view clustering problem takes $\mathcal{T}$ and multiple clustering results $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ as input and outputs D. The goodness of D is measured by an evaluation measure defined in Section 3.2 and by evidence that the user can discover useful knowledge.

2.2 Related Work

Methods in information visualization can be classified into pattern visualization such as decision-tree visualizer of MineSet [20] and data visualization, which includes VisuMClust. A data visualization method can display raw data such as raw value in the input but can also display transformed data such as an agglomerated value for several raw values. WebCANVAS [5] employs a clustering algorithm which learns a mixture of first-order Markov models and visualizes clusters as navigation patterns on a Web site using membership probabilities of clusters. PrototypeLines [18] employs a clustering algorithm which learns a mixture of multinomial distributions models and visualizes time-series data using an information-theoretic criterion for allocating colors to clusters. These methods are effective but are limited because they rely on a single clustering result.

Ever since the birth of association rule discovery [2], attempts for obtaining pontentially useful patterns from transactional data have been made. Examples of such patterns include large itemsets, free itemsets, and closed itemsets but are prohibitely large in number to be employed in visualization. Recently Afrati

et al. proposed to approximate transactional data with a specified number of itemsets and have suggested their use of visualization [1]. Though we think the idea is interesting, we believe that such visualization has the same deficiency as the approach based on a single clustering result.

A recent work [16] proposes a visualization tool to detect changes between two clustering results produced by SOM for different time periods. This work underlines the necessity to associate the used colors between the two clustering results for the visual analysis of the similarities and differences. Nevertheless, this work is limited to two clustering results contrary to our method.

3 VisuMClust: Data Visualization with Multiple Clusterings

This section presents our method VisuMClust for visualizing data from multiple clustering results for knowledge discovery.

3.1 Outlines of our Approach

We explain the outline of VisuMClust with a small example. The input to the problem is a transactional data set and multiple clustering results shown in Figure 1. The transactional data set (left part of Figure 1) consists of nine transactions $t_1, t_2, \ldots, t_9$ described by the set of items $\mathcal{I} = \{A, B, \ldots, J\}$. The right part of Figure 1 shows three clustering results $\mathcal{C}_1, \ldots, \mathcal{C}_3$. Each clustering result is composed of several clusters (e.g. three clusters for $\mathcal{C}_1$). For each cluster $c_{i,j}$, the table indicates both the transactions belonging to $c_{i,j}$ and the items describing $c_{i,j}$. For instance, the cluster $c_{1,2}$ contains the transactions t_5, t_6, t_7 and it is described by the items D, E. In the area of transactional clustering, it is common to produce results in term of such associations between transactions and itemsets [17].

Id	Items
1	A J
2	ABC
3	ABC
4	ABC
5	DE
6	DE H
7	A $DEFGH$
8	A FG I J
9	HI

Id	Clusters				
$\mathcal{C}_1$	$(A; t_1, t_2, t_3, t_4, t_7, t_8)$	$(DE; t_5, t_6, t_7)$	$(I; t_8, t_9)$		
$\mathcal{C}_2$	$(AFG; t_7, t_8)$	$(ABC; t_2, t_3, t_4)$	$(DEH; t_6, t_7)$	$(I; t_8, t_9)$	$(J; t_1, t_8)$
$\mathcal{C}_3$	$(ABC; t_2, t_3, t_4)$	$(DE; t_5, t_6, t_7)$	$(I; t_8, t_9)$	$(J; t_1, t_8)$	

Fig. 1. Example of a transactional data set and associated multiple clustering results.

VisuMClust is a generic approach and *any* transactional clustering method (as well crisp as soft) providing clusters with itemsets can be used. In this paper, we use Eclat as a clustering method because it is efficient on large transactional data and it enables natural interpretation as a soft clustering method [8].

Given the input, VisuMClust allocates a color to each cluster. This allocation should guarantee a "consistent" view of the transactional data set through multiple clustering results: similar clusters in different clustering results are preferably allocated the same color or two similar colors (the method

is depicted in Section 3.2). This strategy highlights differences and similarities between the clustering results. The basic idea is to display each transaction as a set of colored rectangles (i.e., rods) each of which is placed in a column based on several clustering results (Figure 2 shows the display result[3] of VisuMClust on our small example). Consequently each row represents a clustering result and is represented as a line of rods which corresponds to the visualisation result of the transactional data. The order of transactions is equivalent to the order in the data set. Each rod is displayed by using the colors of the groups of clusters to which the corresponding transaction belongs (Table 1 details the colors of the groups of clusters). If a transaction belongs to a single group of clusters in a clustering result, then its rod has a single color (e.g. t_1). With soft-clustering, a transaction can belong to several groups of clusters and then it is displayed for each clustering result by using all the colors of the groups of clusters to which the transaction belongs (e.g., the next to the last rod of the lines belongs to two clusters in $\mathcal{C}_1$ and $\mathcal{C}_3$ while three in $\mathcal{C}_2$). We believe that this is a simple and natural way to present the data to the user.

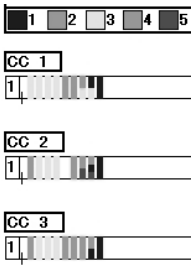

Fig. 2. Example of display result of VisuMClust.

	1 (blue)	2 (green)	3 (yellow)	4 (orange)	5 (red)
Clusters of $\mathcal{C}_1$	$(I; t_8, t_9)$	$(DE; t_5, t_6, t_7)$	$(A; t_2, t_3, t_4, t_7, t_8)$	$\emptyset$	$\emptyset$
Clusters of $\mathcal{C}_2$	$(I; t_8, t_9)$	$(DEH; t_6, t_7)$	$(ABC; t_2, t_3, t_4)$	$(J; t_1, t_8)$	$(AFG; t_7, t_8)$
Clusters of $\mathcal{C}_3$	$(I; t_8, t_9)$	$(DE; t_5, t_6, t_7)$	$(ABC; t_2, t_3, t_4)$	$(J; t_1, t_8)$	$\emptyset$

Table 1. Colors allocated by VisuMClust from the clustering results indicated by Figure 1.

This allocation strategy turned out to be effective in interpretation. In Figure 2, we see that the clusters depicted by the color 3 seem reliable. They gather the same transactions (t_2, t_3, t_4) on $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$. The prototypical transactions of the clusters of color 2 are t_5 and t_6 and, for the color 1, t_9. Such transactions may be essential in interpretating these clusters. The transactions t_7 and t_8 seem difficult to allocate in a set of clusters. They may be outliers or their interpretation may be linked to several sets of clusters. Colors 4 and 5 are under represented and the clusters using them may be unreliable. Obviously, these interpretations are based on hypotheses but we argue that they guide the user in knowledge discovery and Section 4 shows the effectiveness of this approach on data sets.

[3] All the pictures are available in color at http://www.info.unicaen.fr/∼ndurand/visumclust

3.2 Color attribution based on the minimal transversals of a hypergraph

We start by formalizing the problem of allocation of colors. Let k be the maximal number of clusters for a clustering result. The principle is to select one cluster from each clustering result in order to get the set of the most similar clusters and allocate the same color to these clusters. Then, the selected clusters are removed and the process is iterated until no cluster remains. Finally, each cluster is allocated a color and we call a set of clusters having the same color a *group*.

In this work, we apply the Jaccard coefficient as a similarity measure between clusters because it is very usual but other similarity measures can be used. The Jaccard coefficient is the ratio between the number of common and distinct items between two itemsets (i.e., $Jaccard(I_1, I_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|}$) [4]. The similarity of a group is related to the notion of intra-cluster similarity used in clustering [4]. It is the average of the similarities of all pairs of clusters (a cluster is also an itemset) of this group. We call the similarity of a group *intra-color similarity* $SimC$ (a color corresponds to a group). A goodness measure S_D of the allocation of colors to the display result D is the average of the intra-color similarities of all groups. The higher S_D is, the better the allocation of colors is.

However, the set of candidate groups is typically huge in number (there are k^n candidate groups) and a naive method which enumerates all groups and computes their similarities fails. Our idea is to set this problem in the hypergraph area [3] so that we can benefit from results in this domain. We will see that it ensures to select at each step a good group with respect to the intra-color similarity.

A hypergraph can be thought as a generalization of a graph because the edges, called hyperedges, can have more than two vertices. A transversal is a set of vertices meeting all hyperedges. A minimal transversal is a transversal with a minimal number of vertices (i.e., T is a minimal transversal if $\forall\, T' \subset T$, T' is not a transversal). Finding *all* minimal transversals of a hypergraph is a well-studied problem [9] which has many applications including data mining [13].

The task of color attribution can be represented in terms of a hypergraph $\mathcal{H}$ as follows: the vertices represent the clusters and the hyperedges represent the clustering results. By definition of a transversal, the set of all transversals is the set of all candidate groups (e.g., $T' = \{A, AFG, ABC\}$ is a transversal of $\mathcal{H}_1$ from the running example). Thus the minimal transversals are a subset of the candidate groups. An interesting point of the minimal transversals is that they enable to focus on the set of candidate groups gathering identical clusters through the clustering results. For instance, $T = \{A, ABC\}$ and T' are two candidate groups but T is better than T': indeed, with T, the cluster ABC is selected both for $\mathcal{C}_2$ and $\mathcal{C}_3$ (and $Jaccard(ABC, ABC) = 1$ and $SimC(T) = 0.55$) whereas, with T', AFG is picked for $\mathcal{C}_2$ instead of ABC and $Jaccard(ABC, AFG) = 0.2$ and $SimC(T') = 0.29$ (in other terms, T gathers clusters which are more similar than T'). The set of all minimal transversals is the set of all combinations of clusters through the clustering results gathering the identical clusters and VisuMClust uses the minimal transversals as candidate groups. Then, instead

of generating and computing the similarities of all the potential candidate groups, only the candidate groups corresponding to the minimal transversals are used. For $\mathcal{H}_1$, the similarities of the seven minimal transversals ($\{I\}$, $\{DE, DEH\}$, $\{DE, ABC\}$, $\{DE, J\}$, $\{DE, AFG\}$, $\{A, ABC\}$, $\{A, J\}$) are computed whereas there are 60 candidate groups (i.e., the product of the cardinalities of the three clusterings results).

Many algorithms aim at finding all minimal transversals of a hypergraph. In our problem, hypergraphs are dense because clustering results have similarities and common clusters. We have chosen the CMT prototype [14] because it is efficient for such hypergraphs due to its pruning criteria. The sketch of the color attribution algorithm is given below.

Input: $\mathcal{C}_1, ..., \mathcal{C}_n$ where $\mathcal{C}_i = \{c_{i,1}, ..., c_{i,m}\}$.
Output: $D = (color_1, ..., color_p)$ where $color_i = \{c_{1,j}, ..., c_{n,j'}\}$.
Start
0. Color = 0
1. Transform $\mathcal{C}_i$ into a hypergraph $\mathcal{H}$
Repeat step 2-6 until there are no clusters ($c_{i,j}$) without color:
2. Color += 1
3. Compute the minimal transversals of $\mathcal{H}$
4. Select the best candidate $Cand$ (i.e. the higher similarity value)
5. Attribute the current color to each $c_{i,j} \in Cand$
6. Add each $Cand$ to D and remove them from $\mathcal{H}$
End

Specific treatments are performed when few clusters remain to avoid allocating the same color to dissimilar clusters. On the other hand, as a human cannot distinguish too many colors efficiently, VISUMCLUST stops the process after the eighth color has been allocated (if clusters remain, they are colored in black and considered as no relevant).

Table 1 provides the result of the color attribution on the small example. Table 2 indicates the intra-color similarity values for this example.

	blue	green	yellow	orange	red
$SimC$	1	0.77	0.55	0.33	0

Table 2. Intra-color similarity values of each color indicated by Table 1.

	blue	green	yellow	orange	red
$\mathcal{C}_1$	A	DE	I	-	-
$\mathcal{C}_2$	**AFG**	DEH	I	J	**ABC**
$\mathcal{C}_3$	ABC	DE	I	J	-

Table 3. Colors allocated by the baseline method from the clustering results indicated by Figure 1.

We now demonstrate the usefulness of our method for the color allocation by comparing it to a baseline one. The baseline method employs a greedy strategy to build a group. It starts from the first cluster of C_1, then selects the cluster of C_2 which is the most similar to the already selected cluster and iterates until C_n. When at least two clusters have been selected, the intra-color similarity is used to pick the next cluster. Then, the process is iterated for the next color. Contrary to our method, the result of the baseline method depends on the order of the clusters. Table 3 shows the result by using the baseline method on the example from Figure 1. Clearly, the choice of AFG for the color 1 is inappropriate.

4 Experimental Results

The objective of the experiments is twofold. First, we evaluate the improvement brought by the color attribution method of VisuMClust versus the baseline method. Second, we investigate the effectiveness of VisuMClust for the knowledge discovery using a real-world geographical data set.

4.1 Data Sets and General Results

We used five well-known benchmarks[4]: `Mushroom` (8124 transactions, 116 items), `Votes` (435 trans., 32 it.), `Hepatitis` (155 trans., 43 it.), `Ionosphere` (351 trans., 98 it.) and `Titanic` (2201 trans., 8 it.). The geographical real-world database[5] (called `Geo` in this paper) addresses 99 items stemmed from demographic and economic indicators about 69 geographical units (41 English counties and 28 French regions). `Geo` is used by several universities to detect the links between these geographical units.

For each dataset, several clustering results were obtained by running Ecclat with different parameters. Table 4 summarizes the overall characteristics of the results. On `Hepatitis`, `Ionosphere` and `Titanic`, the baseline method creates one additional color. Due to the space limitation, we only provide the display results on `Geo` in Figures 3 and 4.

In all cases, the computation time is negligible for both the minimal transversals or the baseline method and the improvement brought by the minimal transversals method does not increase computation time.

	`Mushroom`	`Votes`	`Hepatitis`	`Ionosphere`	`Titanic`	`Geo`
No. of clusterings	5	3	4	3	4	4
Minimal no. of clusters in a clustering	3	6	3	5	7	3
Maximal no. of clusters in a clustering	5	6	4	6	8	5
No. of colors (VisuMClust)	6	7	4	6	8	5
No. of colors (baseline method)	6	7	5	7	9	5

Table 4. Information about the results.

	`Mushroom`	`Votes`	`Hepatitis`	`Ionosphere`	`Titanic`	`Geo`
Baseline method	0.554	0.617	0.436	0.342	0.675	0.472
VisuMClust	**0.658**	**0.839**	**0.521**	**0.389**	**0.878**	**0.684**
Gain	+18.7%	+35.9%	+19.5%	+13.9%	+30.1%	+44.9%

Table 5. Comparison of the average intra-color similarity.

Table 5 provides the values of the goodness measure S_D for VisuMClust and the baseline method on the data sets. VisuMClust clearly outperforms the baseline method in the quality of the color allocation (average gain = 27.1%).

4.2 Knowledge Discovery on Geographical Real-World Data

We give a qualitative evaluation of VisuMClust on `Geo`. The aim is twofold: evaluate the effectiveness of VisuMClust in knowledge discovery and compare from a qualitative viewpoint our color attribution method to the baseline

[4] *http://www.ics.uci.edu/∼mlearn/* and *http://www.amstat.org/publications/jse/*
[5] *http://atlas−transmanche.certic.unicaen.fr/index.gb.html*

method. Figure 3 presents the display results of VisuMClust on `Geo` by using our color attribution method and Figure 4 with the baseline method. These figures have been shown to geographers, who are experts of these data.

In Figure 3, the colors 1 (blue) and 2 (green) correspond to English and French units, far from `London` and `Paris`, respectively. These units have common features (e.g., demographic indicators expressing a population getting aged). Among these units, some of them (e.g., `Finistre`, `Morbihan` for France and `Cornwall`, `Devon` for England) are also colored in red in the clustering result $\mathcal{C}_4$ because they have also common demographic characteristics (birth rate, age of the population). On the other hand, the capitals and their surrounding areas (like `Reading` for `London` and `Yvelines` for `Paris`) have the same color (3: yellow). This color has captured dynamic units: large agglomerations with good demographic indicators (low death rate, a lot of young people). The color 4 (orange) only concerns English counties near `London`. The difference with respect to the color 3 lies in economic indicators such as industrial level. These observations have been deduced and validated by the geographers.

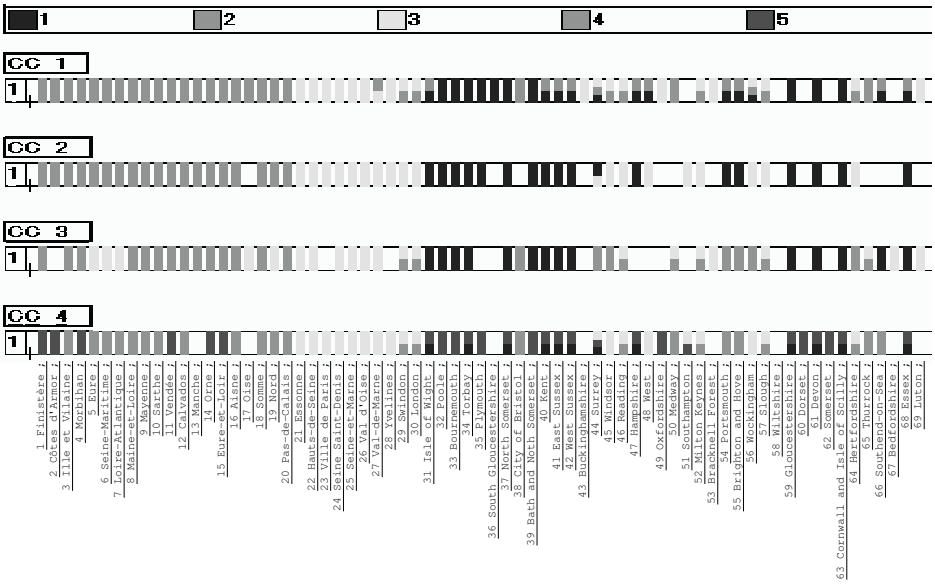

Fig. 3. Results of VisuMClust on the geographical database.

On the contrary, in Figure 4, only one color (2: green) gathers geographical units of one country (England). The other colors mix English and French units. It seems difficult to bring out conclusions because transactions are dispersed and many transactions do not have the same color in all the clusterings (contrary to Figure 3). For instance, the color 3 (yellow) concerns `London`, `Paris` and their surrounding areas ($\mathcal{C}_1$) but also some French regions far from Paris (in $\mathcal{C}_3$ and $\mathcal{C}_4$). There is a contradiction because it is proved that large agglomerations and the French regions from west do not have the same demographic and eco-

nomic features. Similar observations can be done with the color 4 (orange) and some English counties. Compared to Figure 3, only the color 5 gathers the same clusters but it is the last selected color, thus the less reliable.

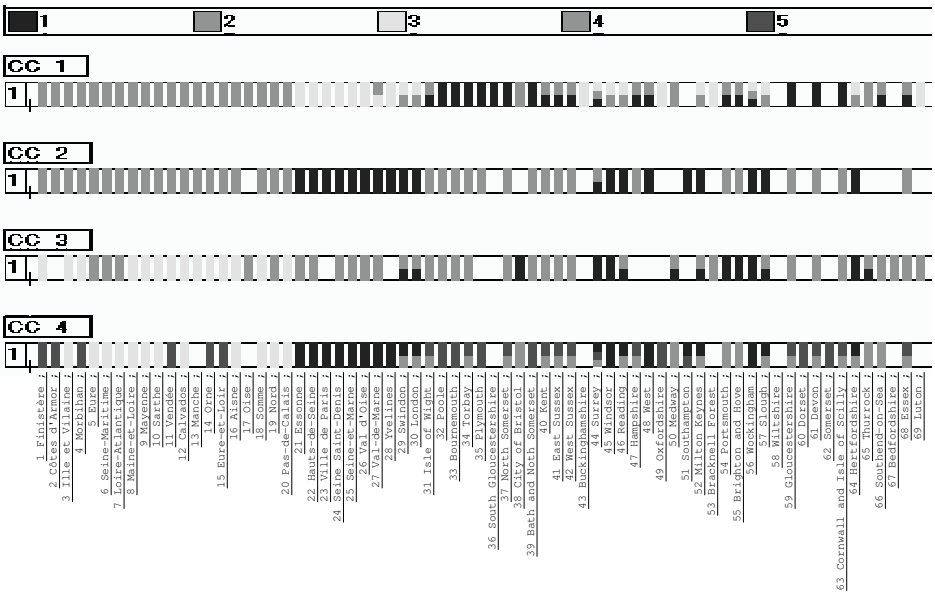

Fig. 4. Results of the baseline method on the geographical database.

Figure 4 did not allow to find valid conclusions contrary to Figure 3 and geographers estimate that the main links between English and French units are better captured in Figure 3.

5 Conclusions

In this paper, we have proposed a multi-view visualization method based on multiple clustering results for transactional data. Conventional visualization methods have a deficiency to be used by a wide range of users because the goodness of a clustering result depends on the user. Our method VisuMClust is expected to overcome this problem by providing a consistent view with its color allocation method based on the minimal transversals of a hypergraph. Objective evaluation for the color attribution and subjective evaluation for knowledge discovery are both promising.

An interesting possibility of VisuMClust is its capability for comparing different clustering results and choosing the best one. Such an application would require various evaluation measures of clustering results and an effective method for handling user feedback. Other issues include automatic selection of interesting transactions to be visualized and ordering of such transactions.

Acknowledgements. The authors thank Céline Hébert, Frédérique Turbout, Arnaud Soulet and François Rioult for stimulating discussions.

References

1. F. N. Afrati, A. Gionis, and H. Mannila. Approximating a Collection of Frequent Sets. In *Proc. 10th Int. Conf. on Knowledge Discovery and Data Mining (KDD'04)*, pages 12–19, Seattle, WA, August 2004.
2. R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Database. *ACM SIGMOD*, 22(2):207–216, May 1993. Washington DC, USA.
3. C. Berge. *Hypergraph.* North Holland, Amsterdam, 1989.
4. P. Berkhin. Survey of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA, 2002.
5. I. V. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. *Data Mining and Knowledge Discovery*, 7(4):399–424, 2003.
6. I. V. Cadez, P. Smyth, and H. Mannila. Probabilistic Modeling of Transaction Data with Applications to Profiling, Visualization, and Prediction. In *Proc. 7th Int. Conf. on Knowledge Discovery and Data Mining (KDD'01)*, pages 37–46, San Francisco, California, USA, August 2001.
7. S. K. Card, J.D. Makinlay, and B. Shneiderman, editors. *Readings in Information Visualization.* Morgan Kaufmann, San Francisco, 1999.
8. N. Durand and B Crémilleux. ECCLAT: a New Approach of Clusters Discovery in Categorical Data. In *Proc. 22nd SGAI Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence (ES'02)*, pages 177–190, Cambridge, UK, 2002.
9. T. Eiter and G. Gottlob. Identifying the Minimal Transversals of a Hypergraph and Related Problems. *SIAM Journal on Computing Archive*, 24(6):1278–1304, 1995.
10. V. Estivill-Castro. Why So Many Clustering Algorithms - A Position Paper. *ACM SIGKDD Explorations*, 4(1):65–75, June 2002.
11. U. Fayyad, G. G. Grinstein, and A. Wierse. *Information Visualization in Data Mining and Knowledge Discovery.* Morgan Kaufmann, San Francisco, 2002.
12. A. Gionis, H. Mannila, and P. Tsaparas. Clustering Aggregation. In *Proc. 21st Int. Conf. on Data Engineering (ICDE05)*, pages 341–352, Tokyo, Japan, April 2005.
13. D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data Mining, Hypergraph Transversals, and Machine Learning. In *Proc. 16th Symposium on Principles of Database Systems (PODS'97)*, pages 209–216, Tucson, Arizona, May 1997.
14. C. Hébert. Enumerating the Minimal Transversals of a Hypergraph Using Galois Connections. Technical report, Univ. Caen Basse-Normandie, France, 2005.
15. J. Hipp, H. Güntzer, and G. Nakhaeizadeh. Algorithms for Association Rule Mining - A General Survey and Comparison. *SIGKDD Explorations*, 2(1):58–64, 2000.
16. D. McG. Squire and D. McG. Squire. Visualization of Cluster Changes by Comparing Self-organizing Maps. In *Proc. 9th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'05)*, pages 410–419, Hanoi, Vietnam, May 2005.
17. R. Pensa, C. Robardet, and J-F. Boulicaut. A Bi-clustering Framework for Categorical Data. In *Proc. 9th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)*, pages 643–650, Porto, Portugal, October 2005.
18. E. Suzuki, T. Watanabe, H. Yokoi, and K. Takabayashi. Detecting Interesting Exceptions from Medical Test Data with Visual Summarization. In *Proc. 3rd IEEE International Conf. on Data Mining (ICDM'03)*, pages 315–322, 2003.
19. E. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, 2001.
20. C. Westphal and T. Blaxton. *Data Mining Solutions.* John Wiley and Sons, New York, 2000.