Towards Constrained Co-clustering in Ordered 0/1 Data Sets

Ruggero G. Pensa, Céline Robardet, and Jean-François Boulicaut

INSA Lyon, LIRIS CNRS UMR 5205
Bâtiment Blaise Pascal
F-69621 Villeurbanne cedex, France
{Ruggero.Pensa, Celine.Robardet, jfboulicaut}@insa-lyon.fr

Abstract. Within 0/1 data, co-clustering provides a collection of bi-clusters, i.e., linked clusters for both objects and Boolean properties. Beside the classical need for grouping quality optimization, one can also use user-defined constraints to capture subjective interestingness aspects and thus to improve bi-cluster relevancy. We consider the case of 0/1 data where at least one dimension is ordered, e.g., objects denotes time points, and we introduce co-clustering constrained by interval constraints. Exploiting such constraints during the intrinsically heuristic clustering process is challenging. We propose one major step in this direction where bi-clusters are computed from collections of local patterns. We provide an experimental validation on two temporal gene expression data sets.

1 Introduction

Many data mining techniques have been designed to support knowledge discovery from 0/1 data, i.e., Boolean matrices whose the rows denote objects and the columns denote Boolean attributes recording object properties.

Table 1. A Boolean context **r**

	g_1	g_2	g_3	g_4	g_5
t_1	1	0	1	1	0
t_2	0	1	0	0	1
t_3	1	0	1	1	0
t_4	0	0	1	1	0
t_5	1	1	0	0	1
t_6	0	1	0	0	1
t_7	0	0	0	0	1

For instance, given **r** in Table 1, object t_1 satisfies only properties g_1 and g_3, and g_4. Exploratory data analysis processes often make use of clustering techniques to get insights about global patterns within the data, i.e., to propose partitions of objects and/or of properties such that a grouping quality measure is

F. Esposito et al. (Eds.): ISMIS 2006, LNAI 4203, pp. 425–434, 2006.

optimized. Many efficient algorithms can provide good partitions but suffer from the lack of explicit cluster characterization. This has motivated the research on conceptual clustering, e.g., the co-clustering approaches [1,2,3,4]. Co-clustering goal is to compute bi-clusters, i.e., associations of (possibly overlapping) sets of objects with sets of properties. An example of an interesting bi-partition in $\mathbf{r}$ is $\{\{\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}\}, \{\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\}\}\}$. The first bi-cluster indicates that the characterization of objects from $\{t_1, t_3, t_4\}$ is that they almost always share properties from $\{g_1, g_3, g_4\}$. Also, properties in $\{g_2, g_5\}$ are characteristic of objects in $\{t_2, t_5, t_6, t_7\}$.

Given a clustering algorithm, the analyst has generally a weak control on the clusters he/she obtains. Typically, he/she can decide for ad-hoc parameter settings which are quite operational and conceptually far from the declarative specification of desired properties. A co-clustering algorithm tries to optimize an objective function (e.g., Goodman-Kruskal's τ coefficient in [1] or the loss of mutual information in [2]) but it might also ensure that some user-defined constraints are satisfied (e.g., the fact that some objects and/or properties have to be together or not). In other terms, we would like to support the search for relevant bi-clusters by enabling user-defined selection predicates (i.e., the conjunction of the objective function optimization constraint with the other user-defined constraints) on bi-partitions as if every possible bi-partition had been computed beforehand. We all know that such a computation is not possible in practice. It explains while a typical (co-)clustering algorithm like Cocluster [2] uses heuristic local optimization to provide a good bi-partition without being able to guarantee the optimal one (i.e., the optimization constraint is relaxed). It explains also that using other user-defined constraints is challenging: enforcing some constraints might lead to lower values for the objective functions. Indeed, to the best of our knowledge, only preliminary approaches have concerned mono-dimensional constrained clustering for simple types of user-defined constraints, mainly the so-called must-link and cannot-link constraints [5,6,7,8].

In this paper, we address the problem of (bi-)cluster discovery when at least one of the dimensions is ordered and when interval constraints are defined w.r.t. orders. One of the typical application domains which motivates our study is temporal gene expression data analysis. In this case, objects denotes gene expression level measurements performed for successive time points on a given organism (e.g., major phases of the developmental cycle), and properties are Boolean gene expression properties (e.g., gene up-regulation or over-expression). For a given organism and during its live cycle, groups of genes are activated and then inhibited, being somehow characteristic of some development stages. A biologist might be interested in finding such co-regulated genes to putatively assign some biological functions and (co-)clustering is a popular techniques for this [4]. In our experience, it is however possible that known stages of a development cycle are not really identified by available clustering algorithms. Bi-clusters may contain samples from different stages, or involve time points which are not contiguous. This is quite confusing for biological interpretation. On another hand, the temporal relationship between the sets of biological conditions can be so strong

that all clustering algorithms return perfect time intervals. In that case, when studying interactions between genes which are co-regulated in different stages, it would be nice to enforce the algorithm to look at alternative bi-partitions, i.e., with no mapping to the development stages.

Therefore, our contribution is twofold. First, we consider constraint-based co-clustering on ordered data and this gives rise to new types of constraints. It is then possible to specify whether a collection of bi-clusters has to be consistent w.r.t. these orders, i.e., the so-called *Interval* and *Non-interval* constraints. It enables to get clusters associated to time intervals or to space regions. Our second contribution concerns the framework which is used to compute the bi-partitions given the specified constraints. We recently proposed a generic framework to compute bi-clusters based on collections of local patterns which capture locally strong associations [9]. From an algorithmic point of view, we show here that it is possible to extend it towards constraint-based bi-cluster mining. A related work in gene expression data analysis is [10]. It provides an algorithm to compute clusters which are constrained by some local patterns maximizing the interclass variance. In such a proposal, some local patterns are used to constrain the partition but they are not selected w.r.t. a declarative specification. Our goal is indeed to build a bi-partition which satisfies user-defined declarative constraints via a preliminary selection of local patterns.

Section 2 provides the problem setting, including the definition of bi-cluster constraints. Section 3 recalls the framework from [9] and discusses its extension towards constrained-based co-clustering. Section 4 concerns our experimental validation on real gene expression data sets. Section 5 concludes.

2 Problem Setting

Assume a set of objects $\mathcal{T} = \{t_1, \ldots, t_m\}$ and a set of Boolean properties $\mathcal{G} = \{g_1, \ldots, g_n\}$. The Boolean context to be mined is $\mathbf{r} \subseteq \mathcal{T} \times \mathcal{G}$, where $r_{ij} = 1$ if property g_j is satisfied by object t_i. For the sake of clarity, $\mathcal{D}$ denotes either $\mathcal{T}$ or $\mathcal{G}$. We define the co-clustering task as follows: we want to compute a partition $P^{\mathcal{T}}$ of K clusters of objects (say $\{P_1^{\mathcal{T}}, \ldots, P_K^{\mathcal{T}}\}$) and a partition $P^{\mathcal{G}}$ of K clusters of properties (say $\{P_1^{\mathcal{G}}, \ldots, P_K^{\mathcal{G}}\}$) with a bijective mapping denoted σ between both partitions s.t. each cluster of objects is characterized by a single cluster of properties ($\sigma : P^{\mathcal{T}} \rightarrow P^{\mathcal{G}}$). The computed bi-partition is denoted $\mathcal{P} = \{P_1, \ldots, P_K\}$ s.t. $P_i = (P_i^{\mathcal{T}}, \sigma(P_i^{\mathcal{T}}))$. Assume now that a real value $s(x_i)$ is associated to each element $x_i \in \mathcal{D}$, where function $s : \mathcal{D} \rightarrow \mathbb{R}$. For instance, $s(x_i)$ can be a temporal or spatial measure related to x_i. In microarray data, where $\mathcal{T}$ is a set of DNA chips, and $\mathcal{G}$ is a set of genes, $s(t_i)$ might be the sampling time related to the DNA chip t_i. On another hand, $s(g_i)$ might measure the absolute position in the whole DNA sequence (if known). The function s, allows to define an order $\preceq$ on dimension $\mathcal{D}$. We say that $x_i \preceq x_j$ iff $s(x_i) \leq s(x_j)$. For the sake of simplicity, if a function s exists on dimension $\mathcal{D}$, then all its elements x_i are ordered, i.e., $\forall i, j$ s.t. $i < j$, $s(x_i) \leq s(x_j)$. We can now redefine the co-clustering task by taking into account the information about the ordered dimension.

Definition 1 (interval and non-interval constraint). *If an order ($\preceq$) is defined on $\mathcal{D}$, an interval constraint on this dimension, denoted $\mathcal{C}_{int}(\mathcal{D}, \mathcal{P})$, enforces each cluster on $\mathcal{D}$ to be an interval: $\forall k = 1 \dots K$, if $x_i, x_j \in P_k^{\mathcal{D}}$ then $\forall x_l$ s.t. $x_i \preceq x_l \preceq x_j$, $x_l \in P_k^{\mathcal{D}}$. A non-interval constraint denoted $\mathcal{C}_{non-int}(\mathcal{D}, \mathcal{P})$ specifies that clusters on $\mathcal{D}$ should not be intervals: $\forall k = 1 \dots K$, $\exists x_i, x_j \in P_k^{\mathcal{D}}$, $\exists x_l \in \mathcal{D}$ s.t. $x_i \preceq x_l \preceq x_j$, $x_l \notin P_k^{\mathcal{D}}$.*

An interval constraints can be used to find clusters which, for instance, contain only adjacent time points (i.e., which are continuous intervals), while a non-interval constraints can be used to find clusters which are not intervals. In the first case, we are able to capture associations which characterize any single stage of the sampling period, while in the second case, we might point out interactions which are somehow time-independent. Other interesting constraints might be defined like extended cannot-link or must-link constraints. Due to space limitation, this is out of the scope of this paper.

3 A Local-to-Global (L2G) Approach

In [9], a generic co-clustering framework is introduced and we have to recall it before its extension towards constraint-based co-clustering. The main idea is to compute bi-partitions from bi-sets which capture locally strong associations between sets of objects and sets of properties. Formally, a bi-set is an element $b_j = (T_j, G_j)$ ($T_j \subseteq \mathcal{T}$, $G_j \subseteq \mathcal{G}$) and we assume that a collection of a priori interesting bi-sets denoted $\mathcal{B}$ has been extracted from $\mathbf{r}$ beforehand. Let us now describe b_j by the Boolean vector $< \mathbf{t}_j >, < \mathbf{g}_j > = < t_{j1}, \dots, t_{jm} >, < g_{j1}, \dots, g_{jn} >$ where $t_{jk} = 1$ if $t_k \in T_j$ (0 otherwise) and $g_{jk} = 1$ if $g_k \in G_j$ (0 otherwise). We are looking for K clusters of bi-sets $\{P_1^{\mathcal{B}}, \dots, P_K^{\mathcal{B}}\}$ ($P_i^{\mathcal{B}} \subseteq \mathcal{B}$). Let us define the centroid of a cluster of bi-sets $P_i^{\mathcal{B}}$ as $\mu_i = < \tau_i >, < \gamma_i > = < \tau_{i1}, \dots, \tau_{im} >$, $< \gamma_{i1}, \dots, \gamma_{in} >$ where τ and γ are the usual centroid components:

$$\tau_{ik} = \frac{1}{|P_i^{\mathcal{B}}|} \sum_{b_j \in P_i^{\mathcal{B}}} t_{jk}, \quad \gamma_{ik} = \frac{1}{|P_i^{\mathcal{B}}|} \sum_{b_j \in P_i^{\mathcal{B}}} g_{jk}$$

We now define our distance between a bi-set and a centroid:

$$d(b_j, \mu_i) = \frac{1}{2} \left(\frac{|\mathbf{t}_j \cup \boldsymbol{\tau}_i| - |\mathbf{t}_j \cap \boldsymbol{\tau}_i|}{|\mathbf{t}_j \cup \boldsymbol{\tau}_i|} + \frac{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i| - |\mathbf{g}_j \cap \boldsymbol{\gamma}_i|}{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i|} \right)$$

It is the mean of the weighted symmetrical differences of the set components. We assume $|\mathbf{t}_j \cap \boldsymbol{\tau}_i| = \sum_{k=1}^{m} a_k \frac{t_{jk} + \tau_{ik}}{2}$ and $|\mathbf{t}_j \cup \boldsymbol{\tau}_i| = \sum_{k=1}^{m} \frac{t_{jk} + \tau_{ik}}{2}$ where $a_k = 1$ if $t_{jk} \cdot \tau_{ik} \neq 0$, 0 otherwise. Intuitively, the intersection is equal to the mean between the number of common objects and the sum of their centroid weights. The union is the mean between the number of objects and the sum of their centroid weights. These measures are defined similarly on properties.

Objects t_j (resp. properties g_j) are assigned to one of the K clusters (denoted i) for which τ_{ij} (resp. γ_{ij}) is maximum. We can enable that a number

Table 2. CDK-MEANS pseudo-code

CDK-MEANS ($\mathbf{r}$ is a Boolean context, $\mathcal{B}$ is a collection of bi-sets in $\mathbf{r}$, K is the number of clusters, MI is the maximal iteration number.)

1. Let $\mu_1 \dots \mu_K$ be the initial cluster centroids. $it := 0$.
2. Repeat
   (a) For each bi-set $b_i \in \mathcal{B}$, assign it to cluster $P_k^{\mathcal{B}}$ s.t. $d(b_i, \mu_k)$ is minimal.
   (b) For each cluster $P_i^{\mathcal{B}}$, compute $\boldsymbol{\tau_i}$ and $\boldsymbol{\gamma_i}$.
   (c) $it := it + 1$.
3. Until centroids are unchanged or $it = MI$.
4. For each $t_j \in \mathcal{T}$ (resp. $g_j \in \mathcal{G}$), assign it to the first cluster $P_i^{\mathcal{T}}$ (resp. $P_i^{\mathcal{G}}$) s.t. τ_{ij} (resp. γ_{ij}) is max.
5. Return $\{P_1^{\mathcal{T}} \dots P_K^{\mathcal{T}}\}$ and $\{P_1^{\mathcal{G}} \dots P_K^{\mathcal{G}}\}$

of objects and/or properties belong to more than one cluster by controlling the size of the overlapping part of each cluster. Thanks to our definition of cluster membership determined by the values of $\boldsymbol{\tau}_i$ and $\boldsymbol{\gamma}_i$, we just need to adapt the cluster assignment step given some user-defined thresholds. A simplified algorithm CDK-MEANS from [9] is recalled in Table 2 (for the sake of brevity, we do not consider cluster overlapping). It computes a bi-partition of $\mathbf{r}$ given a collection of bi-sets $\mathcal{B}$ extracted from $\mathbf{r}$ beforehand (e.g., formal concepts). CDK-MEANS can provide the example bi-partition given in Section 1.

Let us now propose a significant extension of the L2G framework when an Interval or Non-interval constraint has been specified. The key idea is that, to compute a bi-partition which satisfies a (global) constraint, we can process a collection of local patterns which do not violate a local counterpart of this constraint. Using such a local level constraint (possibly associated with a propagation strategy), might enable to get efficiently a bi-partition which satisfies the global level one. Notice that given the state-of-the-art in constraint-based mining of bi-sets, quite efficient algorithms can now extract constrained bi-sets. For instance, in our applications, we use D-MINER [11] for computing complete collections of formal concepts which also satisfy various user-defined constraints. It is possible to enforce the same interval and non-interval constraints in the used bi-set collection. However, in the case of Interval constraint, it might be too stringent in practice, while for Non-interval, it will not be selective enough. For this reason, we propose to relax the Interval constraint and strengthen the Non-interval one on bi-sets by introducing two new local constraints.

Definition 2 (max-gap and min-gap constraints). *Given an order on $\mathcal{D}$, a max-gap constraint on this dimension, denoted $\mathcal{C}_{maxgap}(\mathcal{D}, l, b)$, is satisfied iff, for each pair of consecutive elements $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \leq l$. A min-gap constraint, denoted $\mathcal{C}_{mingap}(\mathcal{D}, l, b)$, is satisfied iff, for each pair of consecutive elements $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \geq l$.*

It is straightforward to prove the following property:

Property 1. *The min-gap constraint is anti-monotonic and can be used to efficient pruning.*

The max-gap constraint does not have any monotonicity property w.r.t. to set inclusion. In fact, for a dimension $D = \{x_1, x_2, \ldots, x_n\}$ a max-gap constraint $\mathcal{C}_{maxgap}(D, 1)$, is not satisfied by the set $X_1 = \{x_2, x_3, x_7\}$, but is satisfied by it superset $X_2 = \{x_2, x_3, x_5, x_7\}$, and by its subset $X_0 = \{x_2, x_3\}$. Then it is neither monotonic nor anti-monotonic[1]. The first one (max-gap) is used for the Interval constraint processing. The second one (min-gap) supports the Non-interval constraint processing. Clearly, final constraint satisfaction is not ensured, but the computational behavior is satisfactory (see the experimental section).

4 Experimental Validation

Evaluation Method. A general criterion to evaluate clustering results consists in comparing the computed partition with a "correct" one. It means that data instances are already associated to some correct labels and that one quantifies the agreement between computed labels and correct ones. A popular measure is the Rand index which measures the agreement between two partitions of m elements. If $\mathbf{C} = \{C_1 \ldots C_s\}$ is our clustering structure and $\mathbf{P} = \{P_1 \ldots P_t\}$ is a predefined partition, each pair of data points is either assigned to the same cluster in both partitions or to different ones. Let a be the number of pairs belonging to the same cluster of $\mathbf{C}$ and to the same cluster of $\mathbf{P}$. Let b be the number of pairs whose points belong to different clusters of $\mathbf{C}$ and to different clusters of $\mathbf{P}$. The agreement between $\mathbf{C}$ and $\mathbf{P}$ can be estimated using

$$Rand(\mathbf{C}, \mathbf{P}) = \frac{a + b}{m \cdot (m-1)/2}$$

which takes values between 0 and 1 and is maximized when $s = t$.

We also want to evaluate co-clustering quality by means of an internal criterion. An interesting measure for this purpose is the symmetrical Goodman and Kruskal's τ coefficient [12] which evaluates the proportional reduction in error given by the knowledge of C^o on the prediction of C^p and vice versa. It is evaluated in a contingency table $\mathbf{p}$. Let p_{ij} be the frequency of relations between an object of a cluster C_i^o and a property of a cluster C_j^p, and $p_{i.} = \sum_j p_{ij}$ and $p_{.j} = \sum_i p_{ij}$. The Goodman-Kruskal's τ coefficient, is defined as follows:

$$\tau = \frac{\frac{1}{2}\sum_i \sum_j (p_{ij} - p_{i.}p_{.j})^2 \frac{p_{i.}+p_{.j}}{p_{i.}p_{.j}}}{1 - \frac{1}{2}\sum_i p_{i.}^2 - \frac{1}{2}\sum_j p_{.j}^2}$$

Time Interval Cluster Discovery. We have studied the impact of the Interval constraint in two microarray data sets, malaria and drosophila. The first one

[1] The search space can be pruned by considering specific ordering at candidate generation phase.

[13] concerns the transcriptome of the intraerythrocytic developmental cycle of Plasmodium Falciparum, i.e., a causative agent of human malaria. The data provide the expression profile of 3 719 genes in 46 biological samples. Each sample corresponds to a time point of the developmental cycle: it begins with merozoite invasion of the red blood cells, and it is divided into three main phases, the ring, trophozoite and schizont stages. The second data set is described in [14]. It concerns the gene expression of the Drosophila melanogaster during its life cycle. The expression levels of 3 944 genes are evaluated for 57 sequential time periods divided into embryonic, larval and pupal stages. The numerical gene expression data given in [13] has been discretized by using one of the encoding methods described in [15]: for each gene g, we assigned the Boolean value 1 to those samples whose expression level was greater than X% of its max expression level. X was set to 25% for malaria and 35% for drosophila. The two matrices have been mined for formal concepts by using D-Miner [11].

We applied Cocluster algorithm [2], and the unconstrained version of CDK-Means with $K = 3$ to identify the three developmental stages. Since the initialization of both algorithms is randomized, we average all the measures on 100 executions. We have measured the Rand index w.r.t. to the real partitioning (which has been inferred from the literature), and the Goodman-Kruskal's coefficient to evaluate the bi-partition quality.

There is a significant difference between the two data sets. In malaria, if Cocluster achieves a good Goodman-Kruskal's coefficient, the bi-clusters obtained by CDK-Means are more consistent with the biological knowledge (i.e., the partition has a higher Rand index). On another hand, the number of comparisons is rather high. What we expect here, is that a constrained approach can obtain the same clustering results by using less computing resources. Instead, for Drosophila, both algorithms fail in finding the correct partitioning w.r.t. the available biological knowledge. The number of jumps is in both cases high, while the Rand index is relatively low. In this case we expect to obtain better results with our constrained clustering approach.

We have defined the Interval constraint on the biological condition dimension. Different levels of the max-gap constraint have been applied and we have studied the impact on the final partition by measuring the Rand index and the Goodman-Kruskal's coefficient.

For malaria (graphics are omitted for sake of brevity), for low values of max-gap, we obtain a better agreement w.r.t. to the three developmental stages, while the Goodman-Kruskal's coefficient is not significantly dissimilar to the one obtained without constraints. On another hand, setting a max-gap constraint considerably reduce the size of the collection, then CDK-Means perform faster. As a secondary observation, notice that our definition of max-gap constraint works for open time intervals. By setting an open time interval constraint, we are always able to obtain a circular sequence of intervals (capturing typical developmental life cycles).

For drosophila, the improvements are more obvious. Unconstrained clustering results have shown that good partitions (with a high Goodman-Kruskal's

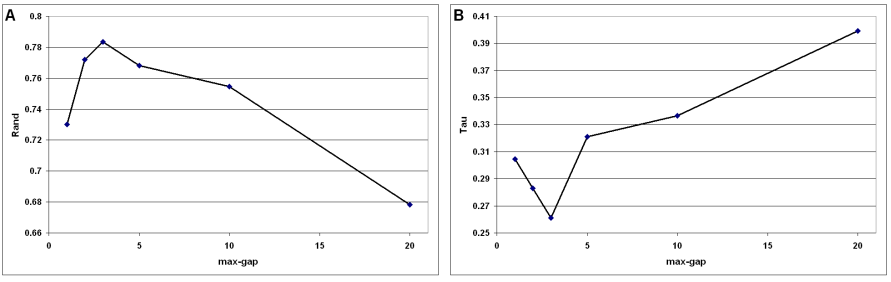

Fig. 1. Results on the drosophila dataset

coefficient) contain a lot of jumps. With a max-gap constraint of 2 or 3, we can sensibly increase the quality of the partition (Fig. 1a) w.r.t. the available biological knowledge. The fact that for these max-gap values, the Goodman-Kruskal's coefficient is minimum (Fig. 1b), indicates that the partition which better satisfies the constraints is not necessarily the "best" one. Moreover, the average number of comparisons is reduced by 60 (max-gap=2) and 30 (max-gap=3).

Using Non-interval Constraint. We have shown how interval constraints can support the discovery of time interval clusters. Within some data (e.g., malaria), an unconstrained approach already gives perfect intervals, and then the question is: is it possible to discover different gene associations which hold between time points belonging to different intervals? To answer this question, we applied the Non-interval constraint to the gene expression data concerning adult time samples of the drosophila melanogaster life cycle. Time samples from t_1 to t_{10} concern the first days of male adult individual life cycle. Time samples from t_{11} to t_{20} concern female individuals.

When we apply CDK-Means (with $k = 2$) without specifying any constraint on this data set, the two intervals $t_1, \ldots, t_{10}$ and $t_{11}, \ldots, t_{20}$ are well identified in the 100 executions of the algorithm. Then, we obtain almost exactly a bi-cluster

Table 3. Clustering results on adult drosophila individuals

		τ		*Rand*	
bi-part.	**inst.**	**mean**	**std.dev**	**mean**	**std.dev**
co:MF	56	0.5605	0.0381	0.82	0.06
co:mixed	44	0.1156	0.0166	0.51	0.02
co:overall	100	0.3648	0.2240	0.69	0.16
cdk:unconst	100	0.4819	0.0594	0.88	0.04
cdk:int	100	0.4609	0.0347	1.00	0.00
cdk:nonint	100	0.1262	0.0761	0.53	0.04

of males and a bi-cluster of females. Moreover , the Goodman-Kruskal's coefficient and the loss in mutual information appears rather stable (see cdk:unconst result on Tab. 3). We computed these coefficients on the 100 bi-partitions returned by COCLUSTER and we noticed a significant unstability (see Tab. 3). It seems that there are two optimum points for which the two measures are distant. For 56 runs, we got a high τ coefficient (mean 0.5605), for the other 44 ones the τ coefficient was sensibly smaller (mean 0.1156). If we consider each group of results separately, the standard deviation is significantly smaller. It means that these two results are two local optima for the COCLUSTER heuristics. From a semantical point of view, the first group of solution reflects the male and female repartition of the individuals, while in the second group each cluster contains both male and female individuals. The average Rand value is 0.69 and the standard deviation is 23% of the mean. Then, we tried to specify a min-gap constraint on the collection of formal concepts. Even for small values of the min-gap constraint, the average Rand value is high, while the standard deviation is lower (12% of the mean for min-gap=2, 4% for min-gap=3) w.r.t. COCLUSTER results. The cdk:nonint row in Tab. 3 summarizes the more stable (w.r.t. the τ coefficient) results obtained with min-gap=10. We also tested whereas an interval constraint could influence the stability of the bi-partition. Setting max-gap=5 enables to get more stable bi-partitions where the Rand index is always equal to one (see cdk:int results in Tab. 3). These results show that, by specifying an Interval or a Non-interval constraint, the user gets some control on the shape of the bi-partition. An algorithm like COCLUSTER has sometimes found bi-clusters where the sex of the individual is the major discriminative parameter. At some moment, it has captured something else. Our thesis is that a biologist might be able to have a kind of supervision on such a process. Moreover, using constraints also speeds up the bi-partition construction because we have to process a reduced collection of bi-sets.

5 Conclusion

Co-clustering is an interesting conceptual clustering approach. Improving bi-cluster relevancy remains a difficult task in real-life exploratory data analysis processes. First, it is hard to capture subjective interestingness aspects, e.g., the analyst's expectation given her/his domain knowledge. Next, when these expectations can be declaratively specified, using them during the computational process is challenging. We have shown that it was possible to use a simple but powerful generic bi-clustering framework based on local patterns. New types of constraints on bi-clusters have been considered when at least one of the dimensions is ordered. Applications on temporal gene expression data analysis have been sketched. Many other applications rely on ordered data analysis and might benefit from such constrained co-clustering approaches. A short-term perspective is to formalize the properties of the global constraints (i.e., constraints on bi-partitions) which can be, more or less automatically, transformed into local level constraints.

Acknowledgements. This research is partially funded by ACI MD 46 Bingo and by EU contract IQ FP6-516169 (FET arm of the IST programme).

References

1. Robardet, C., Feschet, F.: Efficient local search in conceptual clustering. In: Proceedings DS'01. Volume 2226 of LNCS., Washington, USA, Springer-Verlag (2001) 323–335
2. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings ACM SIGKDD 2003, Washington, USA, ACM Press (2003) 89–98
3. Ritschard, G., Zighed, D.A.: Simultaneous row and column partitioning: Evaluation of a heuristic. In: Proceedings of the 14th International Symposium ISMIS 2003. Volume 2871 of LNCS., Maebashi City, Japan, Springer (2003) 468–472
4. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinf. **1**(1) (2004) 24–45
5. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proceedings ICML 2001, Williamstown, USA, Morgan Kaufmann (2001) 577–584
6. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings ICML 2002, Sydney, Australia, Morgan Kaufmann (2002) 307–314
7. Davidson, I., Ravi, S.S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proceedings SIAM SDM 2005, Newport Beach, USA (2005)
8. Davidson, I., Ravi, S.S.: Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In: Proceedings PKDD 2005. Volume 3721 of LNCS., Porto, Portugal, Springer (2005) 59–70
9. Pensa, R.G., Robardet, C., Boulicaut, J.F.: A bi-clustering framework for categorical data. In: Proceedings PKDD 2005. Volume 3721 of LNAI., Porto, Portugal, Springer-Verlag (2005) 643–650
10. Sese, J., Kurokawa, Y., Monden, M., Kato, K., Morishita, S.: Constrained clusters of gene expression profiles with pathological features. Bioinformatics **20**(17) (2004) 3137–3145
11. Besson, J., Robardet, C., Boulicaut, J.F., Rome, S.: Constraint-based concept mining and its application to microarray data analysis. Intelligent Data Analysis **9(1)** (2005) 59–82
12. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classification. Journal of the American Statistical Association **49** (1954) 732–764
13. Bozdech, Z., Llinás, M., Pulliam, B.L., Wong, E., Zhu, J., DeRisi, J.: The transcriptome of the intraerythrocytic developmental cycle of plasmodium falciparum. PLoS Biology **1**(1) (2003) 1–16
14. Arbeitman, M., Furlong, E., Imam, F., Johnson, E., Null, B., Baker, B., Krasnow, M., Scott, M., Davis, R., White, K.: Gene expression during the life cycle of drosophila melanogaster. Science **297** (2002) 2270–2275
15. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.F., Gandrillon, O.: Strong association rule mining for large gene expression data analysis: a case study on human SAGE data. Genome Biology **12** (2002)