A Bi-clustering Framework for Categorical Data

Ruggero G. Pensa[1], Céline Robardet[2], and Jean-François Boulicaut[1]

[1] INSA Lyon, LIRIS CNRS UMR 5205,
F-69621 Villeurbanne cedex, France
[2] INSA Lyon, PRISMa EA INSA-UCBL 2058,
F-69621 Villeurbanne cedex, France
{Ruggero.Pensa, Celine.Robardet, Jean-Francois.Boulicaut}@insa-lyon.fr

Abstract. Bi-clustering is a promising conceptual clustering approach. Within categorical data, it provides a collection of (possibly overlapping) bi-clusters, i.e., linked clusters for both objects and attribute-value pairs. We propose a generic framework for bi-clustering which enables to compute a bi-partition from collections of local patterns which capture locally strong associations between objects and properties. To validate this framework, we have studied in details the instance CDK-MEANS. It is a K-MEANS-like clustering on collections of formal concepts, i.e., connected closed sets on both dimensions. It enables to build bi-partitions with a user control on overlapping between bi-clusters. We provide an experimental validation on many benchmark datasets and discuss the interestingness of the computed bi-partitions.

1 Introduction

Many data mining techniques have been designed to support knowledge discovery from categorical data which can be represented as Boolean matrices: the rows denote objects and the columns denote Boolean attributes that enable to record object properties as attribute-value pairs. For instance, given $\mathbf{r}$ in Table 1, we say that object t_2 satisfies properties g_2 and g_5. Clustering is one of the major data mining tasks and it has been studied extensively, including for the special case of categorical or Boolean data. Its main goal is to identify a partition of objects and/or properties such that an objective function which specifies its quality is optimized [1]. Thanks to local search optimizations, many efficient algorithms can provide good partitions but suffer from the lack of explicit cluster characterization. It has motivated the research on conceptual clustering [2] and bi-clustering [3,4,5] whose 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 $\mathbf{r}$ (Table 1) 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 $\{t_1, t_3, t_4\}$ almost always share properties $\{g_1, g_3, g_4\}$. A major problem is that most of the bi-clustering algorithms compute non overlapping bi-partitions, while in many application domains, it makes sense to have objects and properties belonging to more than one bi-cluster. It motivates more research on the computation of relevant collections of possibly

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

overlapping bi-clusters. Formal concept analysis [6] might be a solution. Informally, a formal concept is a bi-set (T, G) where the set of objects T and the set of properties G form a maximal (combinatorial) rectangle of true values, e.g., $(\{t_1, t_3\}, \{g_1, g_3, g_4\})$ in **r**. Unfortunately, we generally get huge collections of formal concepts which are difficult to interpret by end-users. In our example, $\{\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}\}$ is not a formal concept $((t_4, g_1) \notin \mathbf{r})$ but can be built from $\{\{t_1, t_3\}, \{g_1, g_3, g_4\}\}$ and $\{\{t_1, t_3, t_4\}, \{g_3, g_4\}\}$ which are "similar enough" formal concepts. It provides the intuition of our approach.

The contribution of this paper is twofold. First we propose a new bi-clustering framework which enables to compute bi-partitions by grouping local patterns which capture locally strong associations between objects and properties, i.e., bi-sets which satisfy some user-defined constraints. Various local patterns are candidates for such a process, e.g., frequent sets of properties associated to their supporting set of objects, formal concepts, etc. Secondly, we study one instance of this framework, the CDK-MEANS algorithm, which builds simultaneously linked partitions on objects and properties. More precisely, we apply a K-MEANS-like algorithm to a collection of bi-sets (formal concepts in our experiments). As a result, objects and properties are intrinsically associated to clusters, depending on their weights in the finally computed centroids. Our experimental validation confirms the added-value of CDK-MEANS w.r.t. other (bi-)clustering algorithms.

In Section 2, we set up our clustering framework and we survey related work. Section 3 discusses our experimental validation methodology and it contains many experimental results on various benchmark datasets. A comparison between CDK-MEANS, two bi-clustering algorithms (COCLUSTER [4] and BI-CLUST [3]), and two classical clustering algorithms (K-MEANS and EM [1]) is given. Scalability issues are discussed and Section 4 concludes.

2   Clustering Model

Assume a set of objects $\mathcal{O} = \{t_1, \ldots, t_m\}$ and a set of Boolean properties $\mathcal{P} = \{g_1, \ldots, g_n\}$. The Boolean context to be mined is $\mathbf{r} \subseteq \mathcal{O} \times \mathcal{P}$, where $r_{ij} = 1$ if property g_j is satisfied by object t_i. We define the bi-clustering task as follows: we want to compute a partition of K clusters of objects (say $\{C_1^o \ldots C_K^o\}$) and a partition of K clusters of properties (say $\{C_1^p \ldots C_K^p\}$) with a mapping between both partitions such that each cluster of objects is characterized by a cluster of properties. Our idea is that bi-partitions can be computed from bi-sets and it will

be instantiated later on formal concepts. Formally, a bi-set is an element $b_j = (T_j, G_j)$ $(T_j \subseteq \mathcal{O}, G_j \subseteq \mathcal{P})$ 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 $< \mathbf{t}_j >, < \mathbf{g}_j > = < t_{j1}, \ldots, t_{jm} >, < g_{j1}, \ldots, g_{jn} >$ 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 $\{C_1, \ldots, C_K\}$ $(C_i \subseteq \mathcal{B})$. Let us define the centroid of a cluster of bi-sets C_i as $\mu_i = < \tau_i >, < \gamma_i > = < \tau_{i1}, \ldots, \tau_{im} >,$ $< \gamma_{i1}, \ldots, \gamma_{in} >$ where τ and γ are the usual centroid components:

$$\tau_{ik} = \frac{1}{|C_i|} \sum_{b_j \in C_i} t_{jk}, \quad \gamma_{ik} = \frac{1}{|C_i|} \sum_{b_j \in C_i} g_{jk}$$

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 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. For this purpose, let us introduce parameters δ_o and δ_p in [0,1] to quantify the membership of each element to a cluster. We say that an object t_j belongs to a cluster C_i^o if $\tau_{ij} \geq (1 - \delta_o) \cdot max_i(\tau_{ij})$. Analogously, a property g_j belongs to a cluster C_i^p if $\gamma_{ij} \geq (1 - \delta_p) \cdot max_i(\gamma_{ij})$. Obviously the number of overlapping objects (resp., properties) depends on the distribution of the values of $\boldsymbol{\tau}_i$ (resp. $\boldsymbol{\gamma}_i$). Notice that if overlapping is allowed, $\delta = 0$ does not imply that each object or property is assigned to a single cluster. The choice of a relevant value for δ is clearly application-dependent. When a bi-clustering structure holds in the data, little values of δ are not enough to provide relevant overlapping. On another hand, in noisy contexts, even little values of δ can give rise to significant overlapping zones.

We can now provide details about the studied instance of this framework: a bi-clustering based on formal concepts. Many efficient algorithms have been developed that can extract complete collections of formal concepts under constraints. We use D-MINER [7].

Our instance CDK-MEANS is presented in Table 2. It computes a bi-partition of a dataset $\mathbf{r}$ given a collection of bi-sets $\mathcal{B}$ extracted from $\mathbf{r}$ beforehand (e.g., formal concepts), the desired number of clusters K, the threshold values for δ_o

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, δ_o and δ_p are thresholds values for controlling overlapping)

1. Let $\mu_1 \dots \mu_K$ be the initial cluster centroids. $k := 0$.
2. Repeat
   (a) For each bi-set $c \in \mathcal{B}$, assign it to cluster C s.t. $d(c, \mu_i)$ is minimal.
   (b) For each cluster C_i, compute $\boldsymbol{\tau_i}$ and $\boldsymbol{\gamma_i}$.
   (c) $k := k + 1$.
3. Until centroids are unchanged or $k = MI$.
4. If overlap is allowed, for each $t_j \in \mathcal{O}$ (resp. $g_j \in \mathcal{P}$), assign it to each cluster C_i^o (resp. C_i^p) s.t. $\tau_{ij} \geq (1 - \delta_o) \cdot max_i(\tau_{ij})$ (resp. $\gamma_{ij} \geq (1 - \delta_p) \cdot max_i(\gamma_{ij})$).
5. Else, for each $t_j \in \mathcal{O}$ (resp. $g_i \in \mathcal{P}$), assign it to the first cluster C_i^o (resp. C_i^p) s.t. τ_{ij} (resp. γ_{ij}) is max.
6. Return $\{C_1^o \dots C_K^o\}$ and $\{C_1^p \dots C_K^p\}$

---

and δ_p, and a maximum number of iterations MI. On our example $\mathbf{r}$, CDK-Means provides the bi-partition given in Section 1. The complexity is linear in $\mathcal{B}$ and scalability issues are discussed in Section 3.

Related work. [3] and [4] bi-clustering methods alternatively refine a partition when the other one is fixed, optimizing respectively the Goodman-Kruskal's τ coefficient and the loss in mutual information. The first interesting difference is instead of considering objects and properties as separated entities during the bi-clustering task (even if objective functions are computed on both sets), CDK-MEANS considers their associations as the elements to process. The second one is that CDK-Means can easily compute partitions with overlapping clusters.

3   Experimental Validation

Different techniques can be used to evaluate the quality of a partition. An external criterion 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 Jaccard coefficient [1].

To evaluate the quality of our bi-clustering using an internal criterion we use Goodman and Kruskal's τ coefficient [8]. It is evaluated in a co-occurrence table $\mathbf{p}$ and it discriminates well bi-partitions w.r.t. the intensity of the functional link between both partitions [3]. p_{ij} is the frequency of relations between an object of a cluster C_i^o and a property of a cluster C_j^p. $p_{i.} = \sum_j p_{ij}$ and $p_{.j} = \sum_i p_{ij}$. The $\tau_{\mathcal{Q}}$ coefficient evaluates the proportional reduction in error given by the knowledge of C^o on the prediction of C^p ($\tau_{\mathcal{O}}$ will denote the measure when exchanging the partitions):

$$\tau_{\mathcal{Q}} = \frac{\sum_i \sum_j \frac{p_{ij}^2}{p_{i.}} - \sum_j p_{.j}^2}{1 - \sum_j p_{.j}^2}$$

We report on experiments using eight well-known datasets taken from the UCI ML Repository[1] and from the JSE Data Archive[2]. All the experiments have been performed on a PC with 1 Gb RAM and a 3.0 GHz P4 processor. First, without considering the class variable, we have processed each dataset with D-Miner [7]. Minimal set size constraints have been used for mushroom and credit-a (minimal itemset and objectset sizes $(13, 15)$ and $(6, 15)$) to obtain complete collections of formal concepts before using CDK-Means.

Table 3. Goodman-Kruskal's coefficient values for different bi-clustering algorithms (mr-2 and mr-5 refer to mushroom with 2 and 5 clusters)

Dataset	Dim.	Bi-Clust Max	Cocluster Max	Cocluster Mean	CDK-means Max	CDK-means Mean
voting	435×48	**0.320**	**0.320**	0.315±0.002	0.311	0.311±0.000
titanic	2201×8	**0.332**	0.321	0.226±0.076	0.314	0.160±0.109
iris-2	150×8	0.543	0.543	0.357±0.195	0.543	0.474±0.056
iris-3	150×8	**0.544**	0.390	0.379±0.045	0.523	0.329±0.080
zoo-2	101×16	0.191	0.186	0.157±0.034	**0.192**	0.165±0.020
zoo-7	101×16	-	0.080	0.065±0.009	**0.083**	0.049±0.015
breast-w	699×18	**0.507**	0.507	0.474±0.121	0.498	0.498±0.000
credit-3	690×52	0.104	0.014	0.003±0.003	**0.110**	0.091±0.015
credit-2	690×52	-	0.012	0.006±0.004	**0.096**	0.055±0.011
mr-2	8124×126	-	**0.198**	0.158±0.026	0.176	0.157±0.017
mr-5	8124×126	**0.187**	0.119	0.097±0.009	0.116	0.112±0.004
ads	3279×1555	-	0.006	0.003±0.001	**0.538**	0.137±0.109

We compared CDK-Means bi-partitions with those obtained by Cocluster [4], and Bi-Clust [3]. As the initialization of these algorithms is randomized, we executed them 100 times on each dataset and we selected the result which returned the best Goodman-Kruskal's coefficient. The number of desired clusters for each experiment has been set to the number of class variable values, except for Bi-Clust which automatically determines the number of clusters. Bi-Clust is available within WEKA[3] and we were not able to process internet-ads (more than 1500 properties). We summarize these results in Table 3. We provide only the $\tau_{\mathcal{Q}}$ coefficients. The corresponding $\tau_{\mathcal{O}}$ coefficients are equal or not significantly different. Notice that, when CDK-Means has the worst results, the Goodman-Kruskal's coefficient is not significantly dissimilar from other algorithm coefficients. On the other hand, for internet-ads, the coefficient obtained with CDK-Means is considerably higher than the one obtained with Cocluster. This is due to the high dimension of the dataset which is not well handled by the other algorithms. Also the average behavior is similar to the one of Cocluster. The average values of the two algorithms are often similar, as well

---
[1] http://www.ics.uci.edu/~mlearn/MLRepository.html
[2] http://www.amstat.org/publications/ jse/jse_data_archive.html
[3] http://www.cs.waikato.ac.nz/ml/weka/

Table 4. Jaccard coefficient values w.r.t. class variable for different algorithms

Dataset	Bi-Clust	Cocluster	K-Means	EM	CDK-means
voting	0.6473	0.6473	0.6027	0.6459	**0.6737**
titanic	0.4281	0.4651	0.3697	0.3697	**0.4745**
iris-2	0.4992	0.4992	**0.5117**	0.4992	0.4992
iris-3	0.4932	0.5240	**0.5394**	**0.5394**	0.5144
zoo-2	0.5141	**0.5630**	0.5027	0.5179	0.5141
zoo-7	-	0.1647	0.1843	**0.2325**	0.2212
breast-w	0.8246	0.8287	0.7777	**0.8328**	0.7666
credit-3	0.4233	0.3869	0.3765	0.3405	**0.4452**
credit-2	-	0.4360	0.4698	0.4442	**0.4915**
mr-2	-	0.6819	0.3496	**0.6976**	0.6356
mr-5	**0.5068**	0.3450	0.3192	0.3364	0.3375
ads	-	0.4317	-	-	**0.8019**

as the standard deviation values. Notice that for voting-records and breast-w, CDK-Means has always produced the same bi-partition.

CDK-Means generally needs for more execution time than the other algorithms because it processes possibly large collections of formal concepts. In these benchmarks, the extraction of formal concepts by itself is not that expensive (from 1 to 20 seconds). Using minimal size constraints during the formal concept extraction phase enables to reduce the collection size and it will be discussed later. For titanic, iris, and zoo, CDK-Means performs in less than one second, while for breast-w, credit-a and internet-ads, the average execution time is less than one minute. For mushroom, the average execution time is about seven minutes since more than 50 000 formal concepts have to be processed.

We also used the Jaccard index to compare the agreement of the object partitions with those determined by the class variables. Here again, we have selected the bi-partition with the highest τ_Q coefficient[4]. We provide the comparisons in Table 4. Again, our algorithm is competitive w.r.t. the other bi-clustering methods. With the exception of breast-w, our algorithm always performs as or better than Bi-Clust, and most of the times better than Cocluster.

Finally, we have compared our results, w.r.t. two classical clustering algorithms, the WEKA implementations of K-Means and EM (see Table 4). Except for breast-w, our algorithm is competitive w.r.t the other ones. For most datasets, CDK-Means performs better than K-Means and EM. Once again, when our algorithm obtains the best result, the difference with the score of the others is significant (except on breast-w). These results show that our clustering of formal concepts is a relevant approach for both partitioning and bi-partitioning tasks.

Scalability Issues. Collections of formal concepts are usually huge, especially in intrinsically noisy data. Since CDK-Means has a linear complexity in the number of bi-sets, it can be time-consuming. An obvious solution is to select some formal concepts, for instance the ones which involve enough objects and/or properties. Interestingly, such minimal size constraints can be pushed into formal concept mining algorithms [7]. Not only it enables the extraction in hard

---

[4] Clearly, it does not lead to the highest Jaccard's index.

Table 5. Clustering results on ads-internet with different minimal size constraints

(σ_p,σ_o)	$|\mathcal{B}|$	time(s)	τ(mean)	τ(max)	J-class	J-ref
(0,0)	7682	33	0.137 ± 0.109	0.538	0.8019	1
(4,4)	2926	8	0.194 ± 0.137	0.565	0.6763	0.6737
(5,5)	2075	5	0.254 ± 0.148	0.565	0.6862	0.7490
(5,10)	1166	2.5	0.223 ± 0.119	0.511	0.6745	0.7405
(7,10)	873	2	0.204 ± 0.095	0.549	0.6172	0.6658
(10,10)	586	1.5	0.227 ± 0.125	0.543	0.6080	0.7167

contexts, but also, intuitively, it removes formal concepts which might be due to noise. We therefore guess that this can increase the quality of the clustering result. Let σ_o be the minimal size of the object set and σ_p be the minimal size of the property set. Properties (resp. objects) that are in relation with less than σ_o objects (resp. σ_p properties) will not be included in any formal concept. As our bi-partitioning method is based only on a post-processing of these patterns, these objects and/or properties can not be included in the final bi-partition. This is not necessarily a problem if we prefer a better robustness to noise. However, one can be interested in finding a bi-partition that includes all objects and properties. An obvious solution is to add the top and bottom formal concepts $(\mathcal{O},\emptyset)$ and $(\emptyset,\mathcal{P})$. This has been done in some experiments (mushroom, credit-a) and we noticed that the decrease of the Jaccard and Goodman-Kruskal's co-efficients were not significant. We made further experiments to understand the impact of using minimal size constraints on both the execution time and the quality of the computed bi-partition. We have considered internet-ads as the most suitable for these experiments (high cardinality for both object and property sets). We extracted formal concepts by setting some combinations of constraints ($0 \leq \sigma_p < 10$ and $0 \leq \sigma_o < 10$) and by adding $(\mathcal{O},\emptyset)$ and $(\emptyset,\mathcal{P})$.The results are summarized in Table 5. It shows that, increasing the minimal size threshold considerably reduces the number of extracted formal concepts and thus the average execution time. Also the extraction time decreases from 4 seconds (for $\sigma_p = \sigma_o = 0$) to one second (for $\sigma_p = \sigma_o = 10$). Moreover, the maximum Goodman-Kruskal's coefficient does not change significantly. In some cases it is greater than the coefficient computed when no size constraint is used. Also the average values of the Goodman-Kruskal's measures are better in general (while standard deviation values are similar). We then computed the Jaccard index of the different partitions w.r.t. the class variable (J-class column) and the partition obtained without setting any constraint (J-ref column). The slight variability of the Jaccard indexes and the high values of the τ measures show that they are still consistent w.r.t. the class one. Finally, results are always better than those obtained by using COCLUSTER (see Fig. 3 and Fig. 4) whose average execution time is about 4.2 seconds. In other terms, increasing σ_p and σ_o can eliminate the impact of noise due to sparse sub-matrices. In particular, grouping larger formal concepts can improve the relevancy of bi-partitions. Notice that if we do not add $(\mathcal{O},\emptyset)$ and $(\emptyset,\mathcal{P})$, we get better results involving a subset of the original matrix: constraints can be triggered to trade-off between the coverage of the bi-partition and the quality of the result.

4   Conclusion and Future Work

We have introduced a new bi-clustering framework which exploits local patterns in the data when computing a collection of (possibly overlapping) bi-clusters. The instance CDK-Means builds simultaneously a partition on objects and a partition on properties by applying a K-Means-like algorithm to a collection of extracted formal concepts. Our experimental validation has confirmed the added-value of CDK-Means w.r.t. other (bi-)clustering algorithms. We demonstrated that such a "from local patterns to a relevant global pattern" approach can work. Due to the lack of space, we omitted the experimental results on real data and the study of overlapping clusters [9]. Many other instances of the framework might be studied. For instance, given extracted local patterns, alternative clustering techniques can be considered. Also, other kinds of local patterns (i.e., relevant bi-sets which are not formal concepts) could be considered. Finally, an exciting challenge concerns constraint-based clustering. Our framework gives rise to opportunities for pushing constraints at two different levels, i.e., during local pattern mining but also when building bi-partitions from them.

References

1. Jain, A., Dubes, R.: Algorithms for clustering data. Prentice Hall, Englewood cliffs, New Jersey (1988)
2. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Machine Learning **2** (1987) 139–172
3. Robardet, C., Feschet, F.: Efficient local search in conceptual clustering. In: Proceedings DS'01. Number 2226 in LNCS, Springer-Verlag (2001) 323–335
4. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings ACM SIGKDD 2003, Washington, USA, ACM Press (2003) 89–98
5. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinf. **1** (2004) 24–45
6. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In Rival, I., ed.: Ordered sets. Reidel (1982) 445–470
7. 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
8. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classification. Journal of the American Statistical Association **49** (1954) 732–764
9. Pensa, R.G., Robardet, C., Boulicaut, J.F.: Using locally relevant bi-sets for categorical data conceptual clustering. Research report, LIRIS CNRS UMR 5205 - INSA Lyon, Villeurbanne, France (2005) Submitted to a journal.