WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IJCNN

# A Spectral algorithm for Topographical Co-clustering

Rogovschi Nicoleta
LIPADE, Paris Descartes University
45, rue des Saints Pères
75006 Paris, France
{nicoleta.rogovschi}@parisdescartes.fr

Lazhar Labiod
LIPADE, Paris Descartes University
45, rue des Saints Pères
75006 Paris, France
{lazhar.labiod}@parisdescartes.fr

Mohamed Nadif
LIPADE, Paris Descartes University
45, rue des Saints Pères
75006 Paris, France
{mohamed.nadif}@parisdescartes.fr

***Abstract*—This paper proposes a spectral algorithm for cross-topographic clustering. It leads to a simultaneous clustering on the rows and columns of data matrix, as well as the projection of the clusters on a two-dimensional grid while preserving the topological order of the initial data. The proposed algorithm is based on a spectral decomposition of this data matrix and the definition of a new matrix taking into account the co-clustering problem. The proposed approach has been validated on multiple datasets and the experimental results have shown very promising performance.**

## I. Introduction

Clustering has received a significant amount of attention as an important problem with many applications, and a number of different algorithms and methods have emerged over the years. Although many clustering procedures such as hierarchical clustering and k-means aim to construct an optimal partition of objects or, sometimes, variables, there are other methods, known as block clustering methods, which consider the two sets simultaneously and organize the data into homogeneous blocks. In recent years block clustering, also denoted co-clustering or bi-clustering, has become an important challenge in the context of data mining. For datasets arising in text mining and bioinformatics where the data is represented in a very high dimensional space, clustering both dimensions of data matrix simultaneously is often more desirable than traditional one side clustering. Co-clustering which is a simultaneous clustering of rows and columns of data matrix consists in interlacing row clusterings with column clusterings at each iteration [1], [2]; co-clustering exploits the duality between rows and columns which allows to effectively deal with high dimensional data. The earliest co-clustering formulation called *direct clustering* has been introduced by Hartigan [3], who proposed a greedy algorithm for hierarchical co-clustering. Dhillon [2] developed a spectral co-clustering algorithm on word-document data, the largest several left and right singular vectors of the normalized word-document matrix are computed and then a final clustering step using $k$means is applied to the data projected to the topmost singular vectors. In [4], the authors proposed an information-theoretic co-clustering algorithm that presents a non-negative matrix as an empirical joint probability distribution of two discrete random variables and set co-clustering problem under an optimization problem in information theory. Probabilistic model-based clustering techniques have also shown promising results in several co-clustering situations, the co-clustering of binary and contingency data has been treated by using latent block Bernoulli and Poisson models [5], [6]. The co-clustering implicitly performs an adaptive dimensionality reduction at each iteration, leading to better document clustering accuracy compared to one side clustering methods [2]. Co-clustering is also preferred when there is an association relationship between the data and the features (i.e., the columns and the rows) [7].

Even if the co-clustering problem is not the main objective of nonnegative factorization matrix (NMF), this approach has attracted many authors for data co-clustering and particularly for document clustering [8].

In text mining field, Dhillon [9] has proposed a spectral block clustering method by exploiting the duality between rows (documents) and columns (words). In the analysis of microarray data, where data are often presented as matrices of expression levels of genes under different conditions, block clustering of genes and conditions has been used to overcome the problem of choosing the similarity on the two sets found in conventional clustering methods [10]. The aim of block clustering is to try to summarize this matrix by homogeneous blocks.

A wide variety of procedures have been proposed for finding patterns in data matrices. These procedures differ in the pattern they seek, the types of data they apply to, and the assumptions on which they rest. In particular we should mention the work of [4], [7], [5] who have proposed some algorithms dedicated to different kinds of matrices. The basic idea of these methods consists in making permutations of objects and attributes in order to draw a correspondence structure between these two sets. In Table 1 we illustrate this task on binary data and obviously, the tables on the right are preferable because they are more concise. It clearly appears that we can characterize each cluster of rows by a cluster of columns. The data on which we are focused represent matrices crossing documents words. There are high dimensional data, very sparse, where the number of words is much higher than the number of documents.

In this paper we propose a new approach called TSC (To-

pographical Spectral Co-clustering). The proposed approach combines a spectral decomposition of the data matrix and the application of the SOM algorithm to a new matrix resulting from a spectral decomposition. This latter leads to a new representation data in a low dimensional space which makes our algorithm faster and more efficient compared to conventional approaches (SOM, Spherical K-means). Furthermore, our approach allows to visualize the results of the co-clustering on a two-dimensional map.

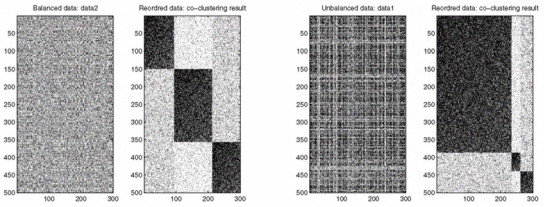


Fig. 1. Left: original data. Right: reorganized data.

The rest of paper is organized as follows. Section 2 introduces the formalism of the traditional Self-Organizing Map (SOM). Section 3 describes our proposed notations and describes the topographical co-clustering model. Section 4 provides details on the TSC algorithm. The results obtained on real document datasets are presented in section 5. Finally, the conclusion summarizes the advantages of our contribution.

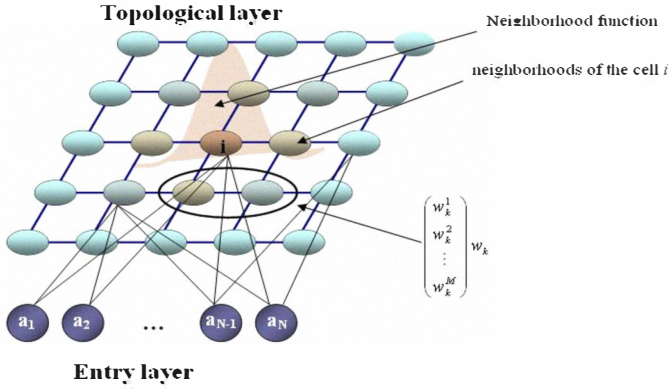


Fig. 2. The principle of the Self-Organizing Map

## II. Traditional Self-Organizing Map (SOM)

The self-organizing maps are increasingly used as tools for data visualization, as they allow projection in low dimensional spaces, typically bi-dimensional. The basic model proposed by Kohonen consists of a discrete set $\mathcal{C}$ of cells called "map". This map has a discrete topology defined by an undirected graph, which usually is a regular grid in two dimensions. For each pair of cells $(r,s)$ on the map, the distance $\delta(r,s)$ is defined as the length of the shortest chain linking cells $r$ and $s$ on the grid. For each cell this distance defines a neighbor cell; in order to control the neighborhood area, we introduce a kernel positive function $\mathcal{K}$ ($\mathcal{K} \geq 0$ and $\lim_{|y|\to\infty} \mathcal{K}(y) = 0$). We define the mutual influence of two cells $r$ and $s$ by $\mathcal{K}_{r,s}$. In practice, as for traditional topological maps we use a smooth function to control the size of the neighborhood as $\mathcal{K}_{j,k} = \exp(\frac{-\delta(r,s)}{T})$. Using this kernel function, $T$ becomes a parameter of the model and as in the Kohonen algorithm, we decrease $T$ from an initial value $T_{max}$ to a final value $T_{min}$.

Let $\Re^M$ be the Euclidean data space and $A = \{\mathbf{a}_i; i = 1,\ldots,N\}$ a set of observations, where each observation $\mathbf{a}_i = (a_i^1, a_i^2, ..., a_i^M)$ is a vector in $\Re^M$. For each cell $k$ of the grid (map), we associate a referent vector (prototype) $\mathbf{w}_k = (w_k^1, w_k^2, ..., w_k^M)$ which characterizes one cluster associated to $k$. We denote by $\mathcal{W} = \{\mathbf{w}_k; \mathbf{w}_k \in \Re^M\}_{k=1}^{|\mathcal{W}|}$ the set of the referent vectors. The structure of the SOM map is presented in figure 2. The set of parameter $\mathcal{W}$ has to be estimated iteratively by minimizing the classical cost function defined as follows:

$$R(\varphi, \mathcal{W}) = \sum_{i=1}^{N} \sum_{k=1}^{|\mathcal{W}|} \mathcal{K}_{k,\varphi(\mathbf{a}_i)} \|\mathbf{a}_i - \mathbf{w}_k\|^2 \quad (1)$$

where $\varphi$ assigns each observation $\mathbf{a}_i$ to a single cell in the map $\mathcal{C}$. This cost function can be minimized using both stochastic or batch techniques [11].

## III. Topographical Co-clustering

This approach is an extension of the SOM algorithm proposed in [11] to the simultaneous topographical clustering of rows and columns of a data matrix. Like for the classical model of self-organizing maps SOM, we use for the TSC approach a neural network with an input layer for the objects (individuals and attributes) and a map $\mathcal{C}$ which contains an topological order like output, the topology oh the map is defined by an undirected graph.

The TSC model uses also the vector quantification, each cell on the map which is the index of the prototype of the searched quantification will be represented by a vector of the same size as the objects. The quantification is done by an assignment function $\varphi$ and the choice of prototypes and assignment function is done by maximizing an objective function denoted by $\mathcal{J}^T(\varphi, W)$. The maximization should allow firstly, to define prototypes to achieve a conservation of the topology of the data and perform in the other hand a partition of $\mathbf{L} = (I \cup J)$ into homogenous subsets or blocks ($I$ is the set of observations and $J$ is the set of columns). Then, in the rest of this paper, we consider the partitioning of $I$ and $J$ simultaneously into $g \geq 2$ non overlapping clusters.

Let $A$ be a $N$ by $M$ data matrix, $R$ be a $N \times g$ index matrix and $C$ be a $M \times g$ index matrix. The matrices $R$ and $C$ take these forms $R = (R_1|R_2|\ldots|R_g)$ and $C = (C_1|C_2|\ldots|C_g)$ where a column $R_k$ or $C_k$ is defined as follows: $r_{ik} = 1$ if row $i$ belongs to cluster $R_k$ and $r_{ik} = 0$ otherwise, and in the same manner $c_{jk} = 1$ if column $j$ belongs to cluster $C_k$ and $c_{jk} = 0$ otherwise.

Our approach consists of two components :

1) Spectral decomposition of the data matrix: This step consists of a spectral decomposition of a weighted data matrix using an SVD and the construction of a new data matrix $\mathbf{D}$ adapted to the problem of co-clustering.
2) Co-clustering of original data $A$ by applying the SOM algorithm on $\mathbf{D}$.

### A. Spectral decomposition

In the following, the number of clusters $g$ on $I$ and $J$ is assumed fixed. We use the following strategy to address the problem of finding a simultaneous partitioning; Compute the first $(g-1)$ left and right eigenvectors of data matrix $A$ to form a $(g-1)$-dimensional embedding of data into a Euclidean space. Then we use a hard-assignment thanks to SOM on this new space to obtain a simultaneous clustering $R$ and $C$.

Let $A$ be an $N$ by $M$ data matrix, taking $D_r = diag(A\mathbb{1})$ and $D_c = diag(A^t\mathbb{1})$ (where $\mathbb{1}$ is the vector of appropriate dimension which values are equal to 1), the matrix $A$ can be approximated by the $(g-1)$th largest eigenvectors of the scaled matrix

$$\tilde{A} = D_r^{\frac{-1}{2}} A D_c^{\frac{-1}{2}}$$

minus the trivial vectors (corresponding to the largest eigenvalue)

Note that we can rewrite $A$ as

$$A = D_r^{\frac{1}{2}}(D_r^{\frac{-1}{2}} A D_c^{\frac{1}{2}}) D_c^{\frac{1}{2}}.$$

It is well known that the largest eigenvalue of

$$\tilde{A} = D_r^{\frac{-1}{2}} A D_c^{\frac{-1}{2}}$$

is equal to $\lambda_0 = 1$ and the associated left and right eigenvectors are respectively [12],

$$U_0 = \frac{D_r^{\frac{1}{2}}\mathbb{1}}{\sqrt{a_{..}}} \text{ and } V_0 = \frac{D_c^{\frac{1}{2}}\mathbb{1}}{\sqrt{a_{..}}}, \text{ with } a_{..} = \sum_{i,j} a_i^j.$$

Applying the spectral decomposition of the scaled matrix $\tilde{A}$ instead on $A$ directly, leading to

$$A = D_r^{\frac{1}{2}} \sum_{k \geq 0} U_k \lambda_k V_k^t D_c^{\frac{1}{2}}. \tag{2}$$

Substract the trivial eigenvectors corresponding to the largest eigenvalue $\lambda_0 = 1$ give

$$A = \frac{D_r \mathbb{1}\mathbb{1}^t D_c}{a_{..}} + D_r^{\frac{1}{2}} \sum_{k \geq 1} U_k \lambda_k V_k^t D_c^{\frac{1}{2}}. \tag{3}$$

Keeping the $(g-1)$th first eigenvectors, we obtain the following approximation

$$A - \frac{D_r \mathbb{1}\mathbb{1}^t D_c}{a_{..}} \approx \sum_{k=1}^{g-1} \tilde{U}_k \lambda_k \tilde{V}_k^t \tag{4}$$

where

$$\tilde{U}_k = D_r^{\frac{-1}{2}} U_k \text{ and } \tilde{V}_k = D_c^{\frac{-1}{2}} V_k.$$

We can approximate $A$ by

$$\sum_{k=1}^{g-1} \tilde{U}_k \lambda_k \tilde{V}_k^t.$$

The data matrix $A$ is expressed in terms of $(g-1)$th first eigenvectors of the scaled matrix $\tilde{A}$. Then we have a $(N \times (g-1))$ matrix

$$U = [U_1, ..., U_{g-1}]$$

formed by the $(g-1)$ left eigenvectors and a $(M \times (g-1))$ matrix

$$V = [V_1, ..., V_{g-1}]$$

formed by the $(g-1)$ right eigenvectors. We then normalize $U$ into $\tilde{U}$ in which

$$\tilde{U}_k = \frac{D_r^{\frac{1}{2}} U_k}{||D_r^{\frac{1}{2}} U_k||},$$

and $V$ into $\tilde{V}$ in which

$$\tilde{V}_k = \frac{D_c^{\frac{1}{2}} V_k}{||D_c^{\frac{1}{2}} V_k||}.$$

The eigenmatrices $\tilde{U}$ and $\tilde{V}$ can be an input of the SOM algorithm via the following new matrix

$$\mathbf{D} = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix} \tag{5}$$

$\mathbf{D}$ is a rectangular matrix of size $((N+M) \times (g-1))$, it is a superposition of the two matrices $\tilde{U}$ and $\tilde{V}$. The term "individuals(rows)-variables(columns)" is to be taken here with a very broad sense. Indeed, the principle of superposition plays on the lack of distinction between these notions when taking into account data, to be restored at the level of the final solution. The set of objects to cluster is the set $\mathbf{L} = I \cup J$, union of the two sets of departure. In the matrix $\mathbf{D}$, individuals and variables are now playing a similar role, so we will refer to by the single term "object". We thus find the problem of one side clustering, since the problem is again to cluster a set of objects, this time however, the set in question is no longer either $I$ or $J$, but the union of the two sets. The solution is a partition of $\mathbf{L}$, we denote by $\begin{pmatrix} R \\ C \end{pmatrix}$ the corresponding matrix partition. It seeks to bring together, in homogeneous clusters the most similar objects (rows and/or columns).

Analogically with the SOM model, we will give the expression of the objective function of the TSC model:

$$\mathcal{J}^T(\varphi, W) = \sum_{i=1}^{N+M} \sum_{k=1}^{|\mathcal{W}|} \mathcal{K}^T_{(\delta(\varphi(i),\ell))} ||\mathbf{d}_i - \mathbf{w}_k||^2 \tag{6}$$

where $\varphi$ the assignment function is defined by

$$\varphi(i) = arg\min_{k} ||\mathbf{d}_i - \mathbf{w}_k||^2 \quad (7)$$

## IV. Topographical Spectral Co-clustering Algorithm (TSC)

The proposed algorithm called *TSC* begins by computing the first $(g-1)$ eigenvectors ignoring the trivial ones. This algorithm is similar in spirit to the one developed by Dhillon [9]. The algorithm embed the input data into the Euclidean space by eigen-decomposing a suitable affinity matrix and then cluster $\mathbf{D}$ using a geometric clustering algorithm. Hereafter, the pseudo code of the proposed algorithm.

**Algorithm 1** TSC

**Input:** data $A$, number of clusters $g$
**Output:** partition matrices $R$ and $C$
**1.** Form the affinity matrix $A$
**2.** Define $D_r$ and $D_c$ to be the diagonal matrices

$$D_r = diag(A\mathbb{1}) \text{ and } D_c = diag(A^t\mathbb{1})$$

**3.** Find $U$,$V$ the $(g-1)$ left-right largest eigenvectors of

$$\tilde{A} = D_r^{-\frac{1}{2}} A D_c^{-\frac{1}{2}}$$

**4.** From $U$ and $V$, form the matrices $\tilde{U}$, $\tilde{V}$ and

$$\mathbf{D} = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix}$$

**5.** Cluster the rows of $\mathbf{D}$ into $g$ clusters by using SOM
**6.** Assign object $i$ to cluster $R_k$ if and only if the corresponding row $\mathbf{d}_i$ of the matrix $\mathbf{D}$ was assigned to cluster $R_k$ and assign attribute $j$ to cluster $C_k$ if and only if the corresponding row $\mathbf{d}_j$ of the matrix $\mathbf{D}$ was assigned to cluster $C_k$.

The *TSC* algorithm contains two majors components: computing the eigenvectors and executing SOM to partition the rows and columns data. We run SOM on $\mathbf{D}$; each row is a $(g-1)$ vector. Standard SOM with Euclidean distance metric has time complexity $O((N+M)dkt)$, where $(N+M)$ is the number of data points plus the number of attributes, and $t$ is the number of iterations required for SOM to converge. In addition, for the *TSC* algorithm there is the additional complexity for computing the matrix eigenvectors $\mathbf{D}$; for computing the largest eigenvectors using the power method or Lanczos method [13], the running time is $O(N^2M)$ per iteration. Similar to other spectral graph clustering method, the time complexity of *TSC* can be significantly reduced if the affinity matrix $A$ is sparse.

## V. Experimental validations

To evaluate the quality of clustering, we adopt the approach of comparing the results to a "ground truth". We use the clustering accuracy for measuring the clustering results. This is a common approach in the general area of data clustering. In general, the result of clustering is usually assessed on the basis of some external knowledge about how clusters should be structured. This may imply evaluating separation, density, connectedness, and so on. The only way to assess the usefulness of a clustering result is indirect validation, whereby clusters are applied to the solution of a problem and the correctness is evaluated against objective external knowledge. This procedure is defined by [14] as "validating clustering by extrinsic classification", and has been followed in many other studies [15], [16]. We feel that this approach is reasonable one if we don't want to judge clustering results by some cluster validity index, which is nothing but a bias toward some preferred cluster property (e.g., compact, or well separated, or connected). Thus, to adopt this approach we need labelled data sets, where the external (extrinsic) knowledge is the class information provided by labels. Hence, if the TSC finds significant clusters in the data, these will be reflected by the distribution of classes. Therefore we operate a vote step for clusters and compare them to the behavior methods from the literature. The so-called vote step consists in the following. For each cluster $c_k \in \mathcal{C}$:

- Count the number of observation of each class $\ell$ (call it $N_{k\ell}$).
- Count the total number of observation assigned to the cell $k$ (call it $N_k$).
- Compute the proportion of observations of each class (call it $S_{k\ell} = N_{k\ell}/N_k$).
- Assign to the cluster the label of the most represented class as follows: $(\ell^* = \arg\max_l(S_{k\ell})$.

A cluster $k$ for which $S_{k\ell} = 1$ for some class labelled $\ell$ is usually termed a "pure" cluster, and a purity measure can be expressed as the percentage of elements of the assigned class in a cluster. The experimental results are then expressed as the fraction of observations falling in clusters which are labelled with a class different from that of the observation. This quantity is expressed as a percentage and termed "error percentage" (indicated as $Err\%$ in the results). Regarding the evaluation method, we choose not to perform cross-validation or similar procedures, considering that the algorithm is trained in a completely unsupervised manner, and calibration already occurs (in a sense) on an external validation data set, that is the set of class labels. Cross-validation or resampling methods, however, could be very useful to assess the stability of the proposed method, by comparing clustering structures in repeated experiments.

### A. Textual datasets

In order to compare the performances of TSC with other traditional unsupervised clustering algorithms, we use many text datasets, which represent the frequency of words in documents.

We used eight datasets for document clustering. "Classic30", "Classic150", "Classic300", "Classic400" are an extract of Classic3 [9] which contains three classes denoted Medline, Cisi, Cranfield as their original database source. Classic30 consists of 30 random documents described by 1000

words and Classic150 consists of 150 random documents described by 3625 words. Tr11 and TR12 were extracted from the "Cluto toolkit". Finally, NG2 (2 classes of documents), NG5 (5 classes) are a subset of 20-Newsgroup data NG20 and composed by 500 documents described by 2000 words, concerning talk.politics.mideast and talk.politics.misc. A short description of these datasets is presented in the table I. Co-clustering is preferred for applications in high dimensional spaces. Most clustering algorithms do not work efficiently in high dimensional spaces due to the curse of dimensionality. It has been shown that in a high dimensional space, the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions. Co-clustering performs an implicit feature selection and overcome object similarity computation. Co-clustering is preferred to classical clustering methods for applications in high dimensional and particulary when $N << M$.

To compute the quality of the performed clustering we adopted an evaluation approach which uses external knowledge (the class information provided by labels). Thus we use the purity index to evaluate the results of the documents clustering. We compared our method with Spherical k-means [17] and SOM approaches. The table II presents the performances obtained by our method. We observe an improvement of the purity on all the databases.

Our approach consists of two components: 1)Spectral decomposition of the data matrix and 2) co-clustering of objects (documents and words) applying the SOM on the matrix $\mathbf{D}$ obtained from the spectral decomposition. The first stage allows us to reduce considerably the data dimension, which will decrease the computational cost thereafter on the second phase. This fact is confirmed by the run-time results reported in the table III.

TABLE I
DESCRIPTION OF THE DATABASES USED FOR THE EVALUATION.

| Databases | # Documents | # Words | #classes |
|---|---|---|---|
| Classic30 | 30 | 1073 | 3 |
| Classic150 | 150 | 3625 | 3 |
| Classic300 | 300 | 5577 | 3 |
| Classic400 | 400 | 6205 | 3 |
| NG2 | 204 | 5831 | 2 |
| NG5 | 878 | 7453 | 5 |
| Tr11 | 414 | 6424 | 5 |
| Tr12 | 313 | 5799 | 5 |

TABLE II
COMPARISON OF THE PURITY INDEX FOR SPERICAL K-MEANS, SOM AND TSC METHODS

| Purity: % | Size of the map | Spherical k-means | SOM | TSC |
|---|---|---|---|---|
| Classic30 | (2 × 3) | 83.31 | 83.33 | 100 |
| Classic150 | (3 × 3) | 90 | 90.24 | 100 |
| Classic300 | (5 × 6) | 93.66 | 82.4 | 99.3 |
| Classic400 | (5 × 5) | 82 | 85.16 | 98.5 |
| NG2 | (5 × 5) | 84.40 | 64.52 | 71.56 |
| NG5 | (6 × 7) | 56.91 | 67.31 | 82.46 |
| TR11 | (4 × 4) | 53.86 | 55.42 | 68.35 |
| TR12 | (7 × 6) | 58.14 | 54.61 | 66.45 |

TABLE III
COMPARISON OF TIME (IN SECONDS) FOR SOM AND TSC METHODS

| Time | Size of the map | SOM | TSC |
|---|---|---|---|
| Classic30 | (2 × 3) | 15.69 | 0.22 |
| Classic150 | (3 × 3) | 506.68 | 0.39 |
| Classic300 | (5 × 6) | 1658.66 | 1.17 |
| Classic400 | (5 × 5) | 2121.51 | 1.17 |
| NG2 | (5 × 5) | 1873.25 | 2.13 |
| NG5 | (6 × 7) | 4384.64 | 7.58 |
| TR11 | (4 × 4) | 2479.43 | 1.23 |
| TR12 | (7 × 6) | 1881.59 | 1.92 |

### B. Zoo data

To illustrate the visualization of the co-clusters, we propose to apply our algorithm on *Zoo* data. This dataset is taken from [18], it contains 101 animals described with 16 variables. Each animal is labelled 1 to 7 according to its class. We use the names used in original data set, [18]. Our results on this dataset is presented in the figure 3. We can simultaneously visualize animals and variables collected by each cell, and also we obtain a topology of the data which is projected on the map. We observe in bottom right corner of the map the sea animals (cell 15, cell 16). On the up left corner we can observe that the domestic birds and wild birds are grouped in neighborhood cells (cell 1, cell 2). The insect family is represented in the middle of the map (cell 9, 10, 11). The animals of the cells 4 and 8 share close characteristics as "milk", "hair", "toothed","predator". The empty cells (5,6,7) form borders between birds and insects families.

## VI. CONCLUSION

In this paper we proposed an extension of the SOM algorithm to co-clustering of the dyadic data. The proposed approach combines a spectral decomposition of the data matrix and the application of the SOM algorithm to a new matrix defined from the matrices of eigenvectors. This new matrix is well adapted to the co-clustering problem. The experimental results obtained using different real high dimensional and sparse databases, show the effectiveness of our approach in terms of classification accuracy and computation time compared to classical SOM and Spherical k-means methods.


## ACKNOWLEDGMENT

This research was supported by the CLasSel ANR project ANR-08-EMER-002.


## REFERENCES


[1] G. Govaert, "Simultaneous clustering of rows and columns," *Control and Cybernetics*, vol. 24, pp. 437–458, 1995.

[2] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the International Conference ACM SIGKDD*, San Francisco, USA, 2001, pp. 269–274.

[3] J. A. Hartigan, "Direct clustering of a data matrix," *Journal of the American Statistical Association*, pp. 123–129, 1972.

[4] I. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic coclustering," in *Proceedings of KDD'03*. KDD'03, September 2003, pp. 89–98.

| | | | |
|---|---|---|---|
| 'chicken','duck', 'pheasant'... *case 1* 'airborne','domestic' | 'crow','flamingo', 'gull','rhea', 'skimmer', 'skua', 'sparrow', 'swan', 'vulture'... *case 2* 'feathers', 'breathes' | 'fruitbat','hamster','vampire'... *case 3* 'tail', 'backbone' | 'antelope','buffalo','calf', 'deer', 'elephant', 'giraffe','gorilla', 'squirrel','wallaby'... *case 4* 'milk', 'hair', 'catsize' |
| *case 5* | *case 6* | *case 7* | 'bear', 'boar', 'leopard', 'lion', 'puma', 'wolf', 'aardvak'... *case8* 'toothed', 'predator' |
| 'honeybee', 'wasp' *case 9* | 'ladybird' *case 10* | 'flea','gnat','moth','housefly', 'termite', 'tortoise','tuatara' ... *case 11* 'legs' | 'penguin', 'sealion', 'slowworm' *case 12* |
| *case 13* | 'seawasp', 'seasnake', 'piviper', 'stingray', 'scorpion'... *case 14* 'venomous' | 'crab', 'crayfish', 'lobster', 'octopus', 'newt', 'toad', ... *case 15* 'eggs' | 'carp', 'catfish', 'dogfish', 'dolphin', 'haddock', 'herring','sole', 'tuna' ... *case 16* 'aquatic', 'fins' |

Fig. 3. The TSC $4 \times 4$ map representing the animals and the variables for each cell.


[5] G. Govaert and M. Nadif, "Block clustering with bernoulli mixture models: Comparison of different approaches," *Computational Statistics and Data Analysis*, vol. 52, pp. 2333–3245, 2008.

[6] ——, "Latent block model for contingency table," *Communications in Statistics, Theory and Methods*, vol. 39, pp. 416–425, 2010.

[7] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proceedings of KDD'06*. Philadelphia, PA: KDD'06, September 2006, pp. 635–640.

[8] L. Labiod and M. Nadif, "Co-clustering under nonnegative matrix tri-factorization," in *ICONIP (2)*, 2011, pp. 709–717.

[9] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," *ACM SIGKDD International Conference*, pp. 269–274, 2001.

[10] Y. Cheng and G. Church, "Biclustering of expression data," in *ISMB2000, 8th International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103.

[11] T. Kohonen, *Self-organizing Maps*. Springer Berlin, 2001.

[12] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *Journal of Machine Learning Research*, pp. 1963–2001, 2006.

[13] G. H. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *in SIAM J. Numer. Anal.*, p. 205224, 1965.

[14] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[15] S. S. Khan and S. Kant, "Computation of initial modes for k-modes clustering algorithm using evidence accumulation." in *IJCAI*, 2007, pp. 2784–2789.

[16] B. Andreopoulos, A. An, and X. Wang, "Bi-level clustering of mixed categorical and numerical biomedical data." *International Journal of Data Mining and Bioinformatics*, vol. 1, no. 1, pp. 19 – 56, 2006.

[17] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, pp. 143–175, 2001.

[18] A. Asuncion and D. Newman, "UCI machine learning repository," http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.