CASOM: SOM for Contingency Tables and Biplot

Rodolphe Priam*
rpriam@gmail.fr

Abstract - *This article presents a new way of dealing with the self-organizing map methods to visualize by an original way qualitative data or histogram vectors as we can find on the Internet e.g. after the pre-processing of plain text documents. The main difference with other known methods is the nature of the processed matrix: a contingency table. By adding constraints during the learning of a mixture of a discrete distribution which models the noise in classes of documents or rows, we obtain a self-organizing map algorithm named CASOM. We explain the properties of the model: metrics, criteria, links with Correspondence Analysis and mean biplot which help to better interpret results. A more general projection available for self-organizing maps in the dual Euclidian space or columns is also introduced. Then, we present some experiments on a corpus of textual short summaries to illustrate the behavior of the algorithm and to show its interest. The conclusion discusses alternative models and gives perspectives of the contribution.*

Key words - Self-Organizing Map, Expectation-Maximization, Correspondence Analysis, Biplot, Textual data analysis

1 Introduction

Self-organizing maps methods were created by Kohonen in the early 1980's. Roughly speaking, Kohonen maps seek to approximate discretized surfaces to model correlation statistics and summarize data distribution. In practice, it is a K-mean[1] algorithm whose classes are constrained on an imaginary lattice. During the learning process, the centers of the classes are updated as in the K-mean method. But all the centers which are near a given center on the lattice also share the data they belong to. So, this smoothing process allows centers which are neighbors on the lattice, to be near each other in the data space. One of the greatest interests of the SOM method is to generalize the principal planes from the Principal Component Analysis method[2] (PCA) in a non-linear way. Correspondence Analysis[3] (CA), a PCA variant for dealing with contingency tables is a very efficient method to extract a structure from an histogram data cloud. Nevertheless, it needs to evolve and to be scaled to be suitable for the huge databases available nowadays. A few years ago, a method KPCA was presented[4] to deal with such data. As CA is like a PCA with a χ^2 metrics, Self-Organizing Maps[5] was applied to this metrics. This method has not yet been used for textual data and give center vectors as continuous multivariables. Here, we propose an alternative approach by modeling classes with multinomial distributions. This last law is today one of the

*This work was begun in IRISA/TEXMEX Team (Rennes, France) and continued in EPUN (Nantes University, France)

best ways to classify[6] texts and can deal successfully with the very sparse textual matrices. Therefore, it seems judicious to include it in the model. As alternative models, we can cite the works[4, 7] which study a stochastic version of SOM with a χ^2 metrics for categorical data. This method should be adapted to construct our dual projections for SOM. Several parametric models have also been proposed these past years as a categorical version of the Generative Topographic Model[8]. These models are quite complex, hard to estimate for large corpuses and to interpret. Our work uses the original decreasing vicinity of SOM during the learning phase. Another work[9] projects the original Probabilistic Latent Semantic Model in the same way, needing more variables to be estimated. Other methods to project data on a plane exist like the classical Multidimensional Scaling (MDS), e.g. Sammon's maps[10], known for its difficulty to be estimated and for the sometimes confusing interpretation of the proximities on the resulting map.
In the following, first we describe the CASOM model that we propose and gives some justifications. Then a biplot method is presented and extented towards a *dual projection* for SOM methods. Then we report the experimental results performed and finally we draw our conclusions. We will mainly analyze textual data as abstracts from scientific articles available on-line. We call *document* (or *text*) a histogram data vector and *term* (or *word*) the component of a *textual* vector. In a formal way, let us suppose we have a corpus of I documents $\mathcal{D} = \{d_i\}_{i=1}^{i=I}$ where d_i represents the i^{th} document $''m_{i_1}m_{i_2}\cdots m_{i_{|d_i|}}''$, the m_{i_j} are the words of the i^{th} document. From that corpus, a vocabulary of J terms $\mathcal{V} = \{m_j\}_{j=1}^{j=J}$ is extracted. In practice, we select $\mathcal{V}$ by calculating the total frequency of each term and keeping the first J terms because of the Zip law[11]. The document matrix is built counting the occurrences of each word from $\mathcal{V}$. Let us have $N_{ij} = \#\{m_{it} = m_j, t \in [1; |d_i|]\}$, $d_i = (N_{i1}, N_{i2}, \cdots, N_{iJ})^T$, and $N_{i\bullet} = \sum_j N_{ij}$, $N_{\bullet j} = \sum_i N_{ij}$, $N_{\bullet\bullet} = \sum_i \sum_j N_{ij}$. The contingency table is built with the d_i as lines, and m_j as columns.

2 CASOM: Generalized Correspondence Analysis

Our algorithm is based on the Topology Preserving Expectation-Maximization or TPEM from [12] which modifies the Classifying EM[13] (CEM) algorithm. This one is a clustering version of the Expectation-Maximization[14] (EM) algorithm where EM is an algorithm to calculate the local maximum of likelihood with latent random variables. Therefore, it is a SOM-like map built with explicit Gaussian distribution for the classes. The counting vectors d_i are now supposed to be i.i.d. realizations of discrete multidimensional random variables following a multinomial law. The algorithm is a clustering process using a mixture of discrete laws with a fuzzification μ_{ik} of the original binary variables c_{ik} which is one if d_i is in class k and zero else, like in a SOM algorithm. During the learning process, the vicinity is reduced to zero when there is no more neighborhood. TPEM is justified by its authors because the CEM of identical isotropic Gaussian laws is equivalent to a K-mean procedure. The Gaussian class distribution can also be argued by the asymptotic properties[15] of the SOM. Since our current model is different, we seek the underlying metrics and discuss some statistical properties of the algorithm. Here θ is the parameter vector merging all unknown variables $P_{j|k}$ (multinomial parameter component), P_k (mixing component), and μ_{ik}. So, the algorithm is a likelihood maximization of a mixture of multinomial distributions by an EM process with a forced fuzzification of the a posteriori probabilities before the maximization step. It enables lateral links between close centers in the lattice. In practice, the batch algorithm encounters what is called a *dead unit* (class center) when no document is assigned to the corresponding class. Besides, as smoothing decreases, μ_{ik} is binary or almost binary, so that $P_{\bullet|k}$ cannot be estimated because the class is empty. In that case, we do not update its value any longer. Thus, we obtain our CASOM algorithm of self-organizing map. Distribution values are initialized with random values or in a more suitable way, with already organized centers as a grid from some linear factorial method. Generally, to display the final map, one uses the U-matrix[16, 17] which shows the local correlation between the closest neighbor classes. A clustering of the class centers, e.g. hierarchical clustering[18], facilitates browsing on the map by permitting the user to focus on the main themes revealed. For the model presented, we propose the natural criterion by analogy with SOM, replacing Euclidian distance by KL distance, where the binary variable h_{kl} is 1 iff e_k and e_l are neighbor or identical:

$$\mathcal{L}_C(\mathcal{D}|\theta) = \sum_i f_i \sum_k \mu_{ik} \sum_l h_{kl}\, KL(f_{\bullet|i}||P_{\bullet|k})$$

This last criterion is approximately[19] minimized by CASOM, ignoring the Bayesian smoothing and near convergence when the centers are well organized. And, we have an approximate local χ^2 metrics, remembering that of Malahanobis : the distance locally adapts[19] itself to each class center:

$$KL(f_{\bullet|i}||\hat{P}_{\bullet|k}) \approx \frac{1}{2}\sum_j \frac{1}{\hat{P}_{j|k}}\left(f_{j|i} - \hat{P}_{j|k}\right)^2$$

Because of the stochastic fluctuation around the mean value, it can also be shown that this last criterion is distributed as a normal law when $min_i N_i$ grows towards infinite values. Morever, as SOM is a non-linear PCA method, the distance justifies that our model is an approximate generalization of the CA method, as the underlying metrics is near the χ^2 one. We call the method CASOM for CA by SOM. Our method also permits a very specific visualization of a corpus by showing rows and columns of a two-way contingency table. We use mean projections of words and documents by showing a document d_i at the Euclidian coordinates $< s|d_i;\hat{\theta} >$ and a word m_j at the Euclidian coordinates $< s|m_j;\hat{\theta} >$ where we have:

$$\begin{array}{lcl} < s|d_i;\hat{\theta} > & = & \sum_k s_k \hat{P}(k|d_i) \text{ with } \hat{P}(k|d_i) \propto \prod_j \hat{P}_{j|k}^{N_{ij}} \\ < s|m_j;\hat{\theta} > & = & \sum_k s_k \hat{P}(k|m_j) \\ \hat{\mathcal{H}}(m_j) & = & -\sum_k \hat{P}(k|m_j) \log_2 \hat{P}(k|m_j) \end{array}$$

It is clear that we must be careful with multimodal distributions showing documents and words at spurious places. So, we choose to select from the finite vocabulary $\mathcal{V}$ only low entropy $\hat{\mathcal{H}}(m_j)$ terms to limit mistakes, with $\hat{P}(k|m_j) \propto \hat{P}_{j|k}$ (or possibly $(\hat{P}_{j|k})^{\alpha}$ with $\alpha > 1$ to underline modes of the distribution). Some edges can be added between very near class distributions, i.e. with small distances D as $D(\hat{P}(k|\square_1), \hat{P}(k|\square_2))$ where $\square_l \in \mathcal{D} \cup \mathcal{V}$. This biplot is a main difference between the original SOM and CASOM: we are able to interpret term statistics and to make comparisons between documents, classes and terms on the same bidimensional map. For classical SOM methods, where centers are continuous, we propose an alternative. Knowing the fact that the K-means is equivalent to a CEM of a mixture of gaussian law with spherical and identical variance matrices, we can write $P(k|m_j) \propto exp(-\rho \sum_i (x_{ij} - c_{jk})^2)$ which, for a good ρ, and c_k a center in $\mathbb{R}^J$, reveals most of the explained intra-variance.

3 Experimental and empirical results

The projected corpus comes from the summaries of the technical home publications of INRIA (http://www.irisa.fr/bibli/publi/), of the past 10 years. These scientific abstracts cover all the research themes of the INRIA institute: 1) Networks and systems, 2) Software engineering and Symbolic calculus, 3) Human-Machine Interface and 4) Simulation and optimization of complex systems.These abstracts are in two languages, French and English. The factorial planes of the multinomial parameters come from the French version with a vocabulary of 480 words. We project this corpus of 1,961 documents on a 12*10 knot mesh and extract a quick view of its content. For these French summaries, we decide to stop the algorithm before near convergence. We noticed a clear empirical link of the properties between CA and CASOM in the Figure 1. We also project the English version of the abstracts, learning the map until the end of the convergence. We thus obtain a well-trained map with natural clusters. Here, for a vocabulary of 476 words, we retain only 1,955 texts. The size of the textual vectors is near the French one. SOM-like methods index data in *semantic* clusters where they can be retrieved by a user providing a query $d_q = (N_{q1} N_{q2} \cdots N_{qJ})$. A Boolean treatment is for instance the intuitive value $\sum_{j:N_{qj}>0} \hat{P}_{j|k}$. We are able to provide different maps to a user, in the Figure 2 underlying various features of the data map by drawing the values as level lines.We illustrate the browsing property of the model. This corresponds to activation maps with level lines for the sum of probabilities of the multinomial for the queries "*knowledge*" and "*interface*" with the part of the subgraph around each projected word. Any other indicator could be used instead of the probabilities. The figures show how the self-organizing map behaves: it is activated on different zones according to the diverse themes of the corpus. For example, the word *knowledge* is statistically near the word *interface* as is demonstrated by the superposable curves obtained after the query ; and *interface* defines too an other well separated theme as is demonstrated by its bimodal distribution. Finally, the graph of words gives us an easy way to find the most interesting and reliable statistical correlations. This output permits a quick study of the main hidden relations between words in $\mathcal{V}$, less apparent on the whole table in the Figure 3 unless we use a color scale for the frequencies or clustering.

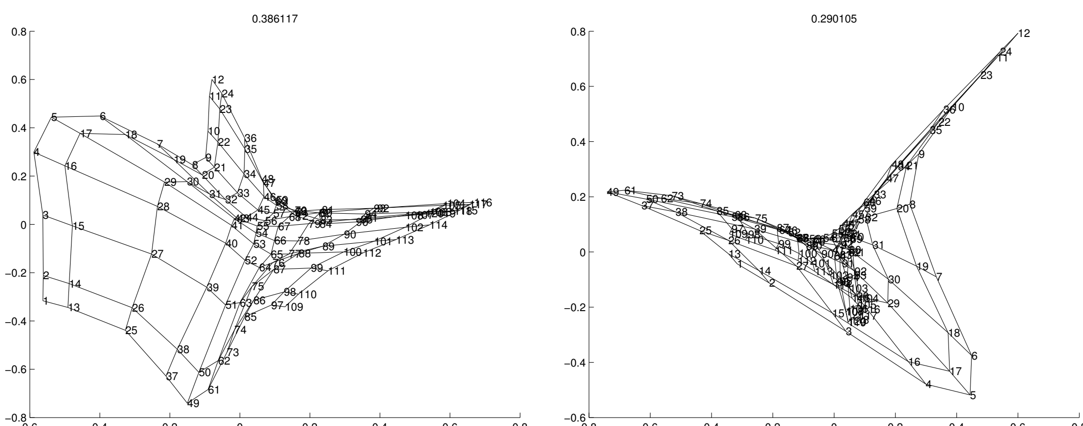

Figure 1: Factorial planes for eigenvectors (1,2) and (2,3). As we stopped before convergence, we get a shape empirically showing the link between CA and CASOM. The values 0.39 and 0.29 are the projected inertia of the corresponding factorial planes. Each knot of the mesh is the class number k.

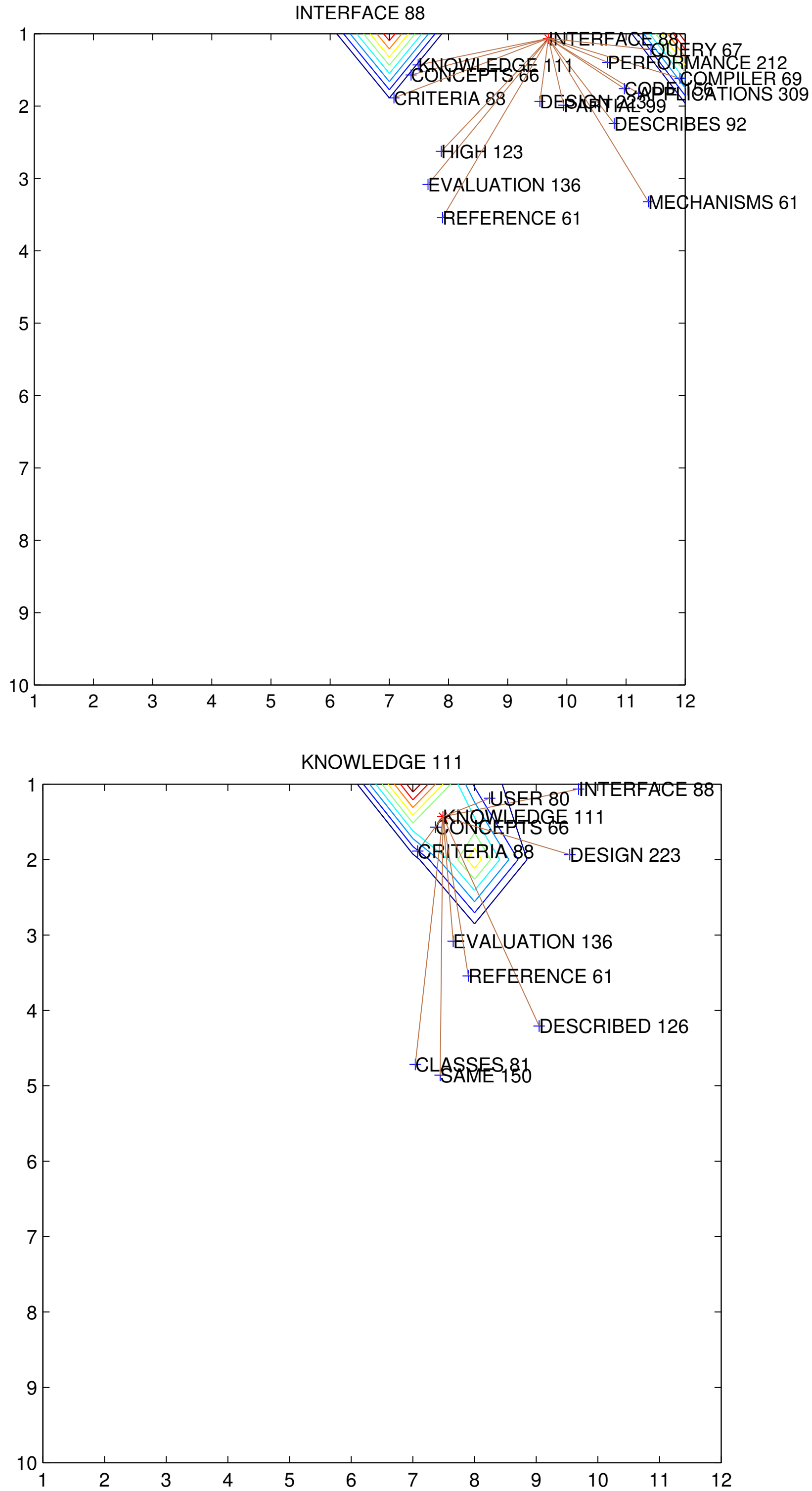

Figure 2: Mean Biplot with graph of words for the two terms *interface*, and *knowledge*. As a remark, every word is written here followed by its total frequency in the corpus. It is shown on the drawings only the restricted graphs of words around the chosen term.

IMAGES	IMAGE	IMAGE	IMAGES	ROBOT	MODELS	KNOWLEDGE	THEIR	MEMORY	DISTRIBUTED	DISTRIBUTED	CODE
IMAGE	MATCHING	IMAGES	SPACE	DYNAMIC	SIMULATION	CRITERIA	DIFFERENT	SIMULATION	MEMORY	APPLICATIONS	APPLICATIONS
SEGMENTATION	IMAGES	BEEN	DEVELOPED	SIGNAL	BOTH	CONTROL	REPORT	SHARED	IMPLEMENTATION	PERFORMANCE	PERFORMANCE
POINTS	IMPORTANT	CLASSIFICATION	OBJECT	SPACE	CONTROL	DESIGN	KNOWLEDGE	PERFORMANCE	PROTOCOL	COMMUNICATION	QUERY
REAL	FEATURES	DEVELOPED	INFORMATION	MOTION	DYNAMIC	METHODS	CONTROL	PARALLEL	APPLICATIONS	NETWORKS	APPLICATION
OBJECTS	PARAMETERS	CONSTRAINTS	BOTH	VISION	BEEN	REPORT	DESIGN	ARCHITECTURES	SHARED	IMPLEMENTATION	IMPLEMENTATION
SURFACE	MODELS	PHASE	BEEN	DETECTION	MATRICES	CONCEPTS	USER	ARCHITECTURE	PERFORMANCE	APPLICATION	DISTRIBUTED
MOTION	REPORT	DIFFERENT	VISION	MODELING	REPORT	THEIR	MAIN	EXECUTION	SIMULATION	PROTOCOL	PROTOCOL
LINES	VISION	MODELS	TECHNIQUE	CRITERIA	MODELING	INTERFACE	ARCHITECTURE	DESIGN	PARALLEL	PROCESS	SOFTWARE
INFORMATION	TOOL	MORE	MORE	TECHNIQUE	ROBOT	EVALUATION	PRESENTS	IMPLEMENTATION	ARCHITECTURE	NETWORK	COMMUNICATION
IMAGES	OBJECT	ESTIMATION	STATISTICAL	DETECTION	BEEN	BEEN	KNOWLEDGE	PARALLEL	DISTRIBUTED	DISTRIBUTED	APPLICATION
CURVES	PARTICULAR	DIFFERENT	OTHER	HAND	THEY	KNOWLEDGE	EACH	MEMORY	PARALLEL	APPLICATIONS	APPLICATIONS
SURFACE	SHAPE	SPACE	REPRESENTATION	MODELS	SIMULATION	PROBLEMS	THEY	DISTRIBUTED	MEMORY	SOFTWARE	LANGUAGE
IMAGE	STRUCTURES	TECHNIQUES	DIFFERENT	PHYSICAL	RESEARCH	MEMORY	INTO	SHARED	PROGRAMMING	COMMUNICATION	MESSAGE
BEEN	WELL	PROPERTIES	INFORMATION	OTHER	SIMULATIONS	STUDIED	SAME	EXECUTION	ENVIRONMENT	APPLICATION	DISTRIBUTED
RESOLUTION	ASSOCIATED	IMAGE	FUNCTION	LARGE	MANY	DESIGN	SEVERAL	MACHINES	DESIGN	PARALLEL	COMMUNICATION
POINTS	OTHER	IMAGES	FINALLY	METHODS	PREVIOUS	TASKS	PARTICULAR	PERFORMANCE	PROGRAMS	PROBLEMS	PROTOCOL
DIFFERENT	REPORT	BEEN	APPROACHES	STRATEGY	STRATEGY	THROUGH	ALLOW	LARGE	EXECUTION	IMPLEMENTATION	IMPLEMENTATION
RECONSTRUCTION	INFORMATION	REPORT	BEING	DESCRIBE	PROPOSE	ASPECTS	MEMORY	TECHNIQUES	COMMUNICATION	PROGRAMMING	DESIGN
CAMERA	LEVEL	STRUCTURE	RESEARCH	STRUCTURES	WHEN	BOTH	PERFORMANCE	IMPLEMENTATION	PROCESSES	REPORT	SUPPORT
IMAGES	MATRIX	LEVEL	MODELS	HAND	CLASSIFICATION	BEEN	EXECUTION	DISTRIBUTED	DISTRIBUTED	DISTRIBUTED	LANGUAGE
MOTION	GEOMETRY	POINT	DETECTION	OTHER	RELATIONS	THEY	PROCESSORS	PARALLEL	PARALLEL	IMPLEMENTATION	ENVIRONMENT
CAMERA	NUMBER	APPROACHES	TECHNIQUE	BOTH	VERY	VERY	SEVERAL	EXECUTION	PROGRAMMING	PARALLEL	SOFTWARE
RECONSTRUCTION	IMAGES	VIEW	METHODS	STRUCTURES	THROUGH	IMPLEMENTATION	EACH	PERFORMANCE	PROGRAMS	REPORT	INFORMATION
IMAGE	METHODS	ESTIMATION	STRUCTURES	CLASSIFICATION	DYNAMIC	BECAUSE	PERFORMANCE	DETECTION	PROCESSES	DYNAMIC	DESIGN
PARAMETERS	DIFFERENT	LINEAR	INFORMATION	LARGE	TREE	DESCRIBE	PARALLEL	PROTOCOL	COMPUTATION	DESIGN	DESCRIPTION
SCENE	LINEAR	REPORT	DISCUSS	STATISTICAL	MOST	IMPORTANT	GLOBAL	GLOBAL	SEQUENTIAL	PROGRAM	DEVELOPMENT
POINTS	STATISTICAL	ABLE	LINEAR	DESIGN	STRUCTURES	POINT	BEEN	CONTROL	PRESENTED	CODE	APPLICATIONS
REAL	REAL	COMPUTER	RANDOM	ASPECTS	CONSTRUCTION	LOCAL	VERY	APPLICATIONS	MECHANISM	LANGUAGES	PROVIDES
MATRIX	OBJECTS	OTHER	MOST	EXACT	CONSIDERED	DESIGN	PROCESSOR	THEIR	LANGUAGE	FRAMEWORK	TOOLS
POINTS	CONSTRAINTS	WITHIN	NUMBER	CLASSIFICATION	SPACE	WORK	PARALLEL	DISTRIBUTED	PROGRAMS	CONTROL	LANGUAGE
CAMERA	POINTS	IMAGE	SEQUENCE	EACH	NUMBER	BEEN	DISTRIBUTED	LOCAL	PARALLEL	LANGUAGE	SPECIFICATION
RECONSTRUCTION	POINT	PROBLEMS	LINEAR	ALLOWS	IMPORTANT	PARALLEL	MEMORY	PROCESS	PROGRAM	EXECUTION	SIGNAL
PARAMETERS	GENERIC	NUMBER	COMPLEXITY	CLASS	CONTEXT	NUMBER	PROCESSORS	THEY	POLICIES	GRAPH	VERIFICATION
EQUATIONS	GIVEN	EFFICIENT	SEQUENCES	MATCHING	VECTOR	IMPORTANT	EXECUTION	EXECUTION	DISTRIBUTED	IMPLEMENTATION	ENVIRONMENT
STRUCTURE	BEEN	VISION	ALLOWS	COMPLEXITY	ENVIRONMENT	EVEN	ONLY	PROGRAMS	PROPERTIES	REPORT	TOOLS
OTHER	ALGEBRAIC	LEVEL	TREE	PROPOSE	SOLUTIONS	SEVERAL	WHEN	DETECTION	SEQUENTIAL	COMPONENTS	FORMAL
MOTION	NUMBER	SMALL	EACH	IMPORTANT	TECHNIQUE	EACH	THERE	PRESENTED	DIFFERENT	ARCHITECTURE	SYNCHRONOUS
SCENE	WHEN	PROCESSING	PROPOSE	EFFICIENT	MOST	DESIGN	TASK	GLOBAL	REPORT	SOFTWARE	PROGRAMMING
PROPOSED	DESCRIBE	DOMAIN	WHEN	DIFFERENT	BEEN	PROCESSING	EACH	APPLICATIONS	INTO	RULES	DESCRIPTION
POINTS	POINTS	SPACE	METHODS	NUMBER	NUMBER	MEMORY	DISTRIBUTED	DISTRIBUTED	DISTRIBUTED	VERIFICATION	LANGUAGE
CONVEX	SIMPLE	ROBOT	DEGREE	IMPLEMENTATION	THAN	NUMBER	NUMBER	EXECUTION	PROPERTIES	LANGUAGE	PROGRAMMING
LINE	GIVEN	POINT	EXACT	CLASSIFICATION	ONLY	PROCESSORS	MESSAGES	PROGRAM	CONTROL	MECHANISM	SPECIFICATION
GIVEN	POINT	RESOLUTION	COMPUTATION	APPLICATION	OPTIMAL	SCHEDULING	SIZE	GLOBAL	GRAPH	PROGRAMMING	LANGUAGES
OBJECTS	SPACE	REPRESENTATION	SEGMENTS	VALUES	VARIABLES	BEEN	STRUCTURE	COMPUTATION	PROGRAM	STRUCTURES	ABSTRACT
POINT	WHERE	LINEAR	IMPLEMENTATION	EACH	BEEN	GENERAL	EXECUTION	STATE	LEVEL	PARALLEL	OBJECT
PROBLEMS	ROBOT	MOST	MOST	PROPOSE	MOST	GIVEN	ONLY	GRAPH	COMPUTATION	STATE	SIGNAL
PLANE	EFFICIENT	EFFICIENT	EACH	CONSISTS	CLASSIFICATION	ONLY	POSSIBLE	GENERAL	EACH	VARIABLES	SEMANTICS
INITIAL	EXACT	GENERAL	MULTIPLE	COMPLEXITY	SEVERAL	POSSIBLE	PROGRAMS	MESSAGES	DETECTION	EXECUTION	STRUCTURES
NUMBER	STRATEGY	FUNCTIONAL	TECHNIQUE	VALUE	SOFTWARE	WHEN	OVER	PROPERTY	OTHER	INPUT	APPLICATIONS
SURFACE	OPTIMAL	REPRESENTATION	POLYNOMIAL	CODES	NUMBER	SCHEDULING	SCHEDULING	SEQUENTIAL	CONSISTENCY	GRAPH	SEMANTICS
MESH	STRATEGY	DIFFERENT	MORE	LINEAR	COMPLEXITY	TASKS	COMMUNICATION	INTO	GRAPH	FORMALISM	LANGUAGE
PART	GIVEN	POLYNOMIAL	THEIR	CODE	PARALLEL	OPTIMAL	TASKS	CONSISTENCY	STATE	OBJECT	PROGRAMMING
EXAMPLES	FINITE	MORE	SOLUTIONS	ONLY	MOST	PARALLEL	PARALLEL	EXECUTION	CLASS	PROGRAMMING	LANGUAGES
PROPOSED	CONSIDER	COMPUTE	ALGEBRAIC	THEIR	WHERE	PROCESSOR	PROCESSORS	TASKS	SEQUENTIAL	EACH	PROGRAM
APPLICATION	CRITERIA	DETECTION	CLASSICAL	EACH	PROBLEMS	COMPLEXITY	CONSIDER	CALLED	ONLY	MORE	ABSTRACT
SURFACES	CLASS	APPROACHES	PROPOSE	GEOMETRIC	POLYNOMIAL	TASK	GENERAL	CONTEXT	CRITERIA	CLASS	LOGIC
GIVEN	PROGRAM	PARTICULAR	DEGREE	GIVEN	CONSIDER	WHERE	WHERE	STRUCTURE	EACH	LANGUAGES	FUNCTIONAL
MESHES	PROGRAMS	VARIOUS	STRUCTURE	THAN	OBTAIN	WHEN	CONSTRAINTS	OPERATIONS	THEORY	PROGRAM	NATURAL
REPORT	REPORT	POINT	METHODS	ALLOWS	WILL	PROBLEMS	TIMES	REPRESENT	BEEN	DIFFERENT	DYNAMIC
FLOW	GIVEN	MATRIX	MATRIX	COMPLEXITY	ONLY	STRUCTURE	GRAPH	GRAPH	CONTROL	STRUCTURES	SEMANTICS
SHAPE	APPLICATIONS	MATRICES	MATRICES	PROBLEMS	SIZE	OPTIMAL	SCHEDULING	RULES	STATE	PROPERTIES	PROGRAM
FIELD	NUMERICAL	ALLOWS	POLYNOMIAL	MATRIX	NUMBER	COMMUNICATION	GENERAL	STRUCTURE	CLASS	THEIR	LANGUAGE
NUMERICAL	METHODS	DEGREE	PROBLEMS	POLYNOMIAL	POLYNOMIAL	SCHEDULING	GRAPHS	EACH	THEORY	NOTION	PROOF
MESH	SIMULATION	METHODS	COMPUTING	GIVEN	TIMES	GRAPH	NUMBER	OBJECTS	CALLED	CONTEXT	PROPERTIES
MESHES	PROBLEMS	FUNCTIONS	PART	MATRICES	LINES	PROCESSOR	TREES	PROCEDURE	GIVEN	ALGEBRAIC	LANGUAGES
FLOWS	REPORT	APPROXIMATION	METHODS	THEORY	LINEAR	DERIVE	TASKS	GLOBAL	OPTIMAL	SEQUENCES	PROGRAMMING
SIMULATION	ALLOWS	TERMS	ALGEBRAIC	FORM	GRAPH	POSSIBLE	PARTICULAR	CALLED	GRAPH	FUNCTIONAL	LOGICAL
EQUATIONS	HERE	POINT	APPLICATIONS	ASYMPTOTIC	NETWORK	DISTRIBUTION	CONSTRUCTION	DISTRIBUTED	RULES	GIVEN	THEIR
COMPUTATION	APPLIED	INTO	NUMERICAL	DEVELOPED	THEIR	MODELS	GENERATED	TREES	STRUCTURES	FUNCTIONS	LOGIC
NUMERICAL	NUMERICAL	APPROXIMATION	FUNCTIONS	PART	MODELS	STOCHASTIC	TIMES	CONSTRUCTION	PROPERTIES	THEIR	PROOF
EQUATIONS	METHODS	SOLUTION	METHODS	METHODS	ASYMPTOTIC	SIMULATION	TREES	TREE	OTHER	FUNCTIONS	LOGIC
FLOW	TERMS	METHODS	LINEAR	PROBLEMS	NETWORKS	NETWORKS	DISTRIBUTED	TREES	THOSE	PROCESSES	CALCULUS
SOLUTION	WHEN	LINEAR	MODELS	DEVELOPED	COMPUTATION	POINT	OPTIMAL	DISTRIBUTED	DEFINE	LANGUAGE	LANGUAGE
DIFFERENT	CONSIDER	MATRICES	MATRICES	ASYMPTOTIC	NUMBER	PROCESSES	PROCESS	PROCESSES	CONSISTENCY	FORM	TERMS
FLOWS	SOLUTION	LARGE	STRUCTURES	FUNCTIONS	THEORY	NUMBER	PROCESSORS	GENERAL	PROVE	MAIN	THEORY
SIMULATION	FLOW	FIELD	WORK	WHERE	STOCHASTIC	AVAILABLE	NUMBER	RELATIONS	OBJECTS	BEHAVIOUR	PROGRAM
SCHEMES	SIMULATION	ILLUSTRATE	FUNCTION	REPORT	STRUCTURES	COMMUNICATION	EACH	EXPLICIT	WHILE	SPECIAL	PROPERTIES
FINITE	NUMBER	WHEN	CONVERGENCE	TIMES	REPRESENT	DISTRIBUTED	PROCESSOR	LIKE	THEIR	THEREFORE	PROVE
PRESENTED	CONVERGENCE	TERMS	MANY	THEORY	PROCESSES	WHEN	TASKS	MEASURES	CLASS	PROPERTIES	RULES
EQUATIONS	PROBLEMS	FUNCTION	FUNCTION	MEASURES	MODELS	NETWORKS	GRAPH	TRANSITION	OPTIMAL	NOTION	LANGUAGE
NUMERICAL	METHODS	CONSTRAINTS	ASYMPTOTIC	MODELS	THEORY	BOUNDS	PROCESSES	SETS	PARTICULAR	PROCESSES	EXTENSION
METHODS	LINEAR	METHODS	COMPUTE	ASYMPTOTIC	PROCESSES	RANDOM	PROVE	INFINITE	CLASS	LOGIC	TYPE
SOLUTION	FIELD	CONDITIONS	WHEN	STATE	PARAMETERS	TREES	STRUCTURE	GRAPH	GRAPH	GIVE	-CALCULUS
MESH	EQUATIONS	CONVERGENCE	GIVE	FUNCTION	GRAPH	NUMBER	NUMBER	FINITE	OTHER	VALUES	TYPES
SCHEME	SOLUTION	FUNCTIONS	FUNCTIONS	FUNCTIONS	RANDOM	WHEN	SIZE	DEFINED	NETS	CONSIDER	TERM
REPORT	CONVERGENCE	LINEAR	PROPOSED	METHODS	NUMBER	GRAPH	GIVEN	COMPUTATIONS	SIMPLE	EXTENSION	LAMBDA
FLOWS	SECOND	WHEN	LINEAR	CLASS	DISTRIBUTION	WHERE	SENSE	GRAPHS	THEIR	CALCULUS	TERMS
FINITE	TECHNIQUES	THEORY	NONLINEAR	PROCESSES	GENERAL	COST	FINITE	TREE	THOSE	PARTICULAR	REDUCTION
PRESENTED	OPTIMIZATION	NUMBER	DIFFERENTIAL	LARGE	MARKOV	POINT	OBTAINED	GENERAL	REDUCTION	INTO	FUNCTIONAL
EQUATIONS	SOLUTION	CONTROL	CONTROL	STATE	RANDOM	TIMES	TREE	NETS	PROCESS	WEAK	CACHE
EQUATION	PROBLEMS	OPTIMAL	FUNCTIONS	CONDITIONS	PROCESS	SERVICE	TIMES	PETRI	POINT	TERMS	LAMBDA
NUMERICAL	EQUATIONS	FUNCTION	FUNCTION	MARKOV	QUEUE	QUEUE	SIZE	TRANSITION	CLASS	TERM	-CALCULUS
FINITE	EQUATION	PROBLEMS	CONTINUOUS	PROCESS	NETWORK	NETWORKS	NUMBER	CLASS	PARTICULAR	PROVE	TYPE
CONDITIONS	CONDITIONS	NONLINEAR	CONDITIONS	UNDER	FINITE	WHEN	WHERE	GRAPH	NUMBER	CLASSICAL	TERMS
SCHEME	CONVERGENCE	CONDITIONS	NETWORK	CONVERGENCE	RATE	NUMBER	PETRI	RESEAUX	SETS	GIVEN	CALCULUS
BOUNDARY	NUMERICAL	SOLUTION	PROPOSED	FINITE	PROBABILITY	STOCHASTIC	RANDOM	FINITE	CITE	SCHEMES	REDUCTION
ELEMENTS	LINEAR	EQUATION	CONVERGENCE	CONSIDER	SERIES	RANDOM	GIVEN	GRAPHS	NETS	TYPE	TYPES
SOLUTION	SOLUTIONS	CONVERGENCE	NONLINEAR	STABILITY	UNDER	PROCESS	DISTRIBUTED	GIVEN	GIVEN	PROPERTIES	THEORY
METHODS	EXISTENCE	SOLUTIONS	FINITE	CONTROL	CONSIDER	PROCESSES	NETS	TREE	INTRODUCE	GIVE	EXPLICIT

Figure 3: Table of the multinomial centers for the English summaries : the terms corresponding to the components with the highest probabilities are shown to caracterize associated classes. It appears that close centers often represent similar themes. We can retrieve here the less apparent two areas where the terms *interface*, and *knowledge* were shown to be the most frequent previously.

4 Conclusion

Our work gives new ideas to deal with self-organizing maps. First, we have presented a new self-organizing map method which has strong links with Correspondence Analysis; as CA is intensively used in textual data analysis, our model is an ideal method to scale CA for large textual datasets where matrices are very sparse with the KL distance known to be competitive, and numerically more efficient than the χ^2 one because zero values are cancelled. We have presented some remarkable properties and the first biplot with a SOM algorithm. Second, new ways to evaluate the quality of the final map have also been briefly given, by visual display or by more quantitative methods. To our knowledge, the parallel between multinomial probability vectors and a discrete bivariate law is an original idea in this domain. Finally, the model is illustrated for KD and IR, providing intuitive indicators. Our paper gives new perspectives for self-organizing map methods in the categorical data analysis field. Biplot is a powerful feature which is lacking in most of the currently developed methods. For instance Multidimensional Scaling could be used to make such a biplot, though inevitably losing the understanding of the projections obtained. Our approach gives tools to have an indepth look at a dataset and also to help as a complementary tool to retrieve data by answering a query. Finally, scaling CASOM to bigger datasets is hopefully possible thanks to the SOM experience[20]. Roughly speaking, our maps can be constructed by any classical SOM process on the reducted data matrix with a joint clustering of the frequencies vector or using[21] fuzzy batch quantities to construct multinomial vectors.

References

[1] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Stat. and Proba.*, 1967, vol. 1, pp. 281–296.

[2] L. Lebart, A. Morineau, and K. Warwick, *Multivariate Descriptive Statistical Analysis*, J. Wiley, 1984.

[3] J. P. Benzecri, *L'analyse des données tome 1 et 2 : l'analyse des correspondances*, Paris:Dunod, 1980.

[4] M. Cottrell, P. Letremy, and E. Roy, "Analysis a contingency table with kohonen maps : a factorial correspondence analysis," *IWANN'93 : 305-311*, 1993.

[5] Teuvo Kohonen, *Self-organizing maps*, Springer, 1997.

[6] Andrew McCallum and Kamal Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*, AAAI Press, Ed., 1998, pp. 41–48.

[7] Smail Ibbou and Marie Cottrell, "Multiple correspondence analysis of a crosstabulation matrix using kohonen algorithm," *ESANN'95*, 1995.

[8] Ata Kaban and Mark Girolami, "A combined latent class and trait model for analysis and visualisation of discrete data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[9] Thomas Hofmann, "Probabilistic topic maps : Navigating throught large text collections," *IDA'99, LNCS 1642, pp 161-172*, 1999.

[10] J.W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 5, no. 18C, pp. 401–409, may 1969.

[11] L. Lebart, A. Salem, and L. Berry, *Explorating textuel data*, Kluwer Academics Publishers, 1998.

[12] C. Ambroise and G. Govaert, "Constrained clustering and kohonen self-organizing maps," *Journal of Classification*, vol. 13, no. 2, pp. 299–313, 1996.

[13] G. Celeux, G. Govaert, and Le Chesnay, "Stochastic algorithms for clustering," *Compstat*, 1990.

[14] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum-likelihood from incomplete data via the em algorithm," *J. Royal Statist. Soc. Ser. B., 39*, 1977.

[15] Hujun Yin and N M Allinson, "On the distribution and convergence of the feature space in self-organising maps," *Neural Computation*, vol. 7, no. 6, pp. 1178–1187, 1995.

[16] A. Ultsch, "New approaches in classification and data analysis. Integration of neural networks with symbolic knowledge processing," *Springer Verlag*, pp. 445–454, 1994.

[17] A. Ultsch and C. Vetter, "Self-organizing feature maps versus statistical clustering : A benchmark," *Research Report No. 9*, 1994.

[18] Juha Vesanto and Esa Alhonieni, "Clustering of the self-organizing map," *IEEE Neural Networks*, vol. 3, no. 11, 2000.

[19] Priam Rodolphe, "CASOM : un SOM pour tableau de contingence (in french)," *to appear in Revue des Nouvelles Technologies de l'Information (numéro spécial) - 18 pages*, 2005.

[20] T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, J. Honkela, and V. Paatero et A. Saarela, "Self organization of a massive document collection," *IEEE Transactions on Neural Networks*, vol. 11, pp. 574–585, 2000.

[21] Priam Rodolphe and Pascale Kuntz, "The CASOM'biplot and Sammon'map," in *COMPSTAT'2004*, 2004.