Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

T. GALLIAT[†], W. HUISINGA[† ‡], AND P. DEUFLHARD[† §]

Self-Organizing Maps Combined with Eigenmode Analysis for Automated Cluster Identification

[†]Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Germany.
Internet: http://www.zib.de/MDGroup
[‡]supported by the Deutsche Forschungsgemeinschaft (DFG) under Grant De 293
[§]Freie Universität Berlin, Fachbereich Mathematik und Informatik, Germany

Preprint SC 99-38 (December 1999)

Self-Organizing Maps Combined with Eigenmode Analysis for Automated Cluster Identification

T. Galliat[1] W. Huisinga[1] P. Deuflhard[1,2]

December 1999

[1] Konrad-Zuse-Zentrum Berlin, Takustr. 7, 14195 Berlin, Germany
[2] Freie Universität Berlin, Fachbereich Mathematik und Informatik, Arnimallee 2–6, 14195 Berlin, Germany

Abstract

One of the important tasks in Data Mining is automated cluster analysis. Self-Organizing Maps (SOMs) introduced by KOHONEN are, in principle, a powerful tool for this task. Up to now, however, its cluster identification part is still open to personal bias. The present paper suggests a new approach towards automated cluster identification based on a combination of SOMs with an eigenmode analysis that has recently been developed by DEUFLHARD ET AL. in the context of molecular conformational dynamics. Details of the algorithm are worked out. Numerical examples from Data Mining and Molecular Dynamics are included.

Keywords. Self-Organizing Maps, cluster analysis, eigenmode analysis, vector quantization, projection methods, nearly uncoupled Markov chains, almost invariant aggregates,chemical conformation dynamics, Data Mining.

1 Introduction

Self-Organizing Maps (SOMs) are a powerful tool to project multi-dimensional data on two-dimensional grids [10, 5]. Compared with traditional vector quantization methods and projection methods, such as Multi Dimensional Scaling (MDS) [2], SOM exhibits two major advantages: (a) it realizes some so-called *topology approximation*, which means that neighbouring objects in the multi-dimensional data space are projected to neighbouring grid points; this allows an interpretation of SOMs as topographic maps of the multi-dimensional data space, (b) it permits to include *additional data* into the treatment during the course of computation [9]. An important field of application of SOMs is Data Mining, especially cluster analysis therein. By using the u-matrix [1] or any other visualization techniques [8], any well-trained analyst will easily identify clusters on a heuristic basis. Different experts will need different computation times and will come up with different results - which gives the whole method an undesirable flavour of personal bias. This is a real obstacle for the use of SOMs in Data Mining, where automated systems are definitely preferred [11].

For this reason, the present paper aims at an *automation* of the cluster identification process. This is done by a proper combination of SOMs with a recently developed eigenmode analysis [4] for a stochastic matrix to be suitably defined. In Section 2 we discuss and assess known approaches to find SOM clusters automatically. In Section 3 we introduce the new dynamically based approach to automatic cluster identification. In Section 4 we describe details of a first already rather efficient algorithm that serves the purpose. Finally, in Section 5, two illustrative examples are given, one from molecular dynamics (a moderate size RNA molecule) and one from Data Mining (a health insurance problem).

2 Known Approaches for Automatic Cluster Identification

In the following, we consider a q–dimensional input space Ω and a two-dimensional SOM formed by a $n \times m$ grid with hexagonal structure and k codebook vectors $W_i \in \Omega$. For each codebook vector z_i denotes the (x, y) position of the related neuron on the grid. For the computation of the map, we use a problem adopted distance function $\mathrm{dist} : \Omega \times \Omega \to \mathbf{R}$ and a Gaussian neighbourhood function $\exp\left(-\frac{\|z_i - z_j\|^2}{2r(t)}\right)$ with the euclidean norm $\|\cdot\|$ and a suitable time-dependent radius function $r(t)$.

A simple idea to identify the clusters on the given map automatically is to run a cluster analysis based on the codebook vectors and using the same distance function as for computing the SOM. However, this idea, which seems reasonable, because the codebook vectors are representatives of the original data, implies the following problem: One has to find a cluster algorithm that neither needs the—a priori unknown—number of clusters as an input (as e.g. the well known c–means–algorithm [7]) nor generates too much different suggestions for clustering the codebook vectors. We do not demand an algorithm that always gives a unique solution[1], but an algorithm that generates only the important clustering possibilities, e.g. in contrast to a hierarchical cluster algorithm [3]. Both requirements on the cluster algorithm are necessary for a really automatic cluster identification.

Even if we suppose that we have such an algorithm, there is another problem: We have not used the two-dimensional structure of the SOM at all. But if we neglect the structure, it makes no sense to use a two-step algorithm—first generating the SOM, afterwards clustering the codebook vectors—instead of clustering the original data directly. In this case, we ignore the power of the SOMs, mainly the topology approximation. Therefore we have to develop a special approach, that not only matches the requirements we have described above, but also uses the information, given by the structure of the SOM.

[1] Usually, in real-world problems, there exists no unique best cluster solution.

3 The New Approach

Motivation. The use of cluster analysis in natural and social sciences or in economics usually implies an a priori assumption: One is interested in finding a clustering, such that similar objects belong to the same cluster, while different objects belongs to different clusters, because one supposes that objects who are similar, still stay similar in the future.[2] Without such an a priori assumption, a cluster analysis makes no or at least less sense.

As a conclusion one can make the following assumption: *The more two clusters are different, the less objects will interchange between these clusters in the future.*

Transformation. On this background, it seems reasonable, if we transform the $k \times k$ distance matrix D, which is build by the inter–distances $\mathrm{dist}(W_i, W_j)$ between all k codebook vectors to a stochastic matrix $S = (s_{ij})$, such that each entry s_{ij} can be interpreted as the probability that an object with nearest codebook vector W_i, will belongs to codebook vector W_j in the future.

To achieve this, we first have to compute a *similarity matrix* $A = (a_{ij})$ with

$$a_{ij} \quad = \quad 1 - \frac{d_{ij}}{\max\limits_j d_{ij}}; \qquad i, j = 1, \ldots, k.$$

Note that the entries satisfy $0 \leq a_{ij} \leq 1$. In a second step, we normalize the rows of the similarity matrix A to 1 resulting in a stochastic matrix S:

$$s_{ij} \quad = \quad \frac{a_{ij}}{\pi_i}; \qquad i, j = 1, \ldots, k \tag{1}$$

with positive weights $\pi_i = \sum_{j=1}^{k} a_{ij}$. Due to the above normalization, the entries s_{ij} may be interpreted as *probabilities.*

But until now, we have not considered the two-dimensional structural information of the map. We can achieve this simply by the following approach[3]:

$$a_{ij} \quad = \quad \exp\left(-\frac{\|z_i - z_j\|^2}{2r}\right)\left(1 - \frac{d_{ij}}{\max\limits_j d_{ij}}\right); \qquad i, j = 1, \ldots, k.$$

For r we use the radius of the map: $r = \min(n, m)/2$. So we have not introduced any additional parameter.

It is easy to show that S is *stochastic* and *reversible*, i.e., $\pi_i s_{ij} = \pi_j s_{ji}$ for all i, j. Therefore the matrix S can now be used within the dynamical cluster algorithm that has recently been introduced by Deuflhard et al. [4].

[2] The meaning of future has always to be interpreted in the context of the given problem. The following simple example should illustrate the assumption: The reason for a company to compute a clustering of their customers is the hope, that a new customer behaves essentially like the old customers, which belongs to the same cluster as the new one.

[3] By this, we use the following property of the SOM: The larger the distance of two codebook vectors on the grid, the larger the distance of the related objects in the multi-dimensional room.

It has to be emphasized, that the application of the algorithm is independent of the previous interpretation. It only needs a reversible stochastic matrix S with respect to the weights π_i. In the next section we will describe details of the algorithm.

4 Algorithmic Realization

In this section we present a concept to identify clusters, which exploits special properties of eigenvectors corresponding to a so–called proximity matrix associated with the problem.

In order to introduce the identification method consider the following setting: Given a set of codebook vectors $\{W_1, \ldots, W_k\}$ and a proximity function $p(W_i, W_j) \in [0,1]$ measuring the degree of association between two codebook vectors. It is $p(W_i, W_j) \approx 1$ for strongly related data and $p(W_i, W_j) \approx 0$, if W_i and W_j are only weakly related. We further request the proximity function to be normalized in the sense that

$$\sum_{j=1}^{k} p(W_i, W_j) = 1$$

for all $i = 1, \ldots, k$. This is no restriction, since it can always be achieved by a suitable normalization of the proximity function (see Section 3).

We are interested in decomposing the data into disjoint clusters $C_1, \ldots, C_c$, such that each cluster C_i groups together related elements, while elements of different cluster are mostly unrelated. Let $p(C_i, C_j)$ denote the proximity between the two clusters C_i, C_j defined by an appropriate average value of $p(W, \hat{W})$ for $W \in C_i$ and $\hat{W} \in C_j$. Then we ask for a decomposition into clusters $C_1, \ldots, C_c$, such that

$$p(C_i, C_i) \approx 1 \quad \text{and} \quad p(C_i, C_j) \approx 0, \quad i \neq j. \tag{2}$$

In order to identify the clusters, the method now uses the proximity matrix $P = (p(W_i, W_j))$ defined in term of the proximity function.

In view of Eq. 2 the clustering problem is equivalent to finding a permutation of the codebook vectors $W_1, \ldots, W_s$ such that the permuted proximity matrix P is *as block-diagonal as possible*, in the sense that the average value over off-blockdiagonal entries is much less than the corresponding blockdiagonal value.

For the identification process, we exploit the following two properties of the proximity matrix (for more details see [4]):

1. The matrix P is stochastic, i.e., its entries are non–negative and the sum of each row equals one. As a consequence, the constant vector $(1, \ldots, 1)$ is an eigenvector corresponding to the eigenvalue $\lambda_1 = 1$; all other eigenvalues λ_i are less or equal in modulus, i.e., $|\lambda_i| \leq 1$. Since P is reversible, it is self–adjoint with respect to some weighted scalar product and consequently, all eigenvalues are real.

2. The presence of clusters corresponds to a block structure of the proximity matrix (for a suitable permutation of the codebook vectors) and a splitting

of the spectrum into a cluster of eigenvalues $\lambda_1, \ldots, \lambda_c$ near 1 and the remaining part of the spectrum. The spectral parts are separated by a gap. The number of clusters, blocks in the proximity matrix and eigenvalues near 1 are equal.

It follows from perturbation analysis [4] that the eigenvectors $X_1, \ldots, X_c$ corresponding to the cluster of eigenvalues near 1 are *almost constant on each cluster*, i.e., if W_i and W_j belong to the same cluster, then $X_l(W_i) \approx X_l(W_j)$ for $l = 1, \ldots, c$. Furthermore, the c–tuple of eigenvector components associated with each W_i,

$$W_i \quad \longmapsto \quad (X_1(W_i), \ldots, X_c(W_i))\,,$$

is sufficient to identify the clusters in the case of weak coupling [4]. Each cluster is the collection of codebook vectors W_i with almost identical c–tuple. We have implemented an algorithm, which also copes with larger perturbations in the eigenvector components due to stronger coupling between the clusters, which is based on the *sign–structure*

$$W_i \quad \longmapsto \quad (\mathrm{sign}\,(X_1(W_i))\,, \ldots, \mathrm{sign}\,(X_c(W_i)))\,,$$

associated with each W_i rather than on the c–tuple itself. A detailed description of this part of the algorithm is given in [4].

For identifying SOM clusters, we now use this algorithm with the stochastic matrix S defined in (1) as the proximity matrix P. As shown in the previous section, S satisfies the necessary properties.

By this we generate a descending sorted sequence of eigenvalues. We now search for large gaps[4] in this sequence. Each of this large gaps separates the group of eigenvalues between the gap and the dominant eigenvalue 1 from the remaining part of the spectrum. This group then allows to identify one possible clustering. Usually there are only a few large gaps, so that our algorithm proposes only few different clusterings.

5 Numerical Examples

To order to show, how our approach works in practice, we shortly present two applications: the first one from the field of molecular dynamics, the second one from insurance business.

Example 1: RNA molecule In this example, the task is to cluster 3–dimensional structures of a molecule, a small triribonucleotide. Here, clusters correspond to groups of similar structures, so–called *geometrical conformations*. Each of the 32.000 molecular structures is roughly described by 37 torsion angles. For more details, see [6].

[4] The minimum size of a large gap depends on the concrete problem. Nevertheless we have used the value 0.08 successfully in different applications.

Since this size has been used successfully in the past, we use a 11×9 grid for the SOM[5]. For the computation of the map, we use a log-inverse radius function and the following special distance function between vectors $V = (v_i)$ and $\hat{V} = (\hat{v}_i)$, that considers the cyclical nature of torsion-angles:

$$dist(V, \hat{V}) = \sum_{i=1}^{37} \left((sin(v_i) - sin(\hat{v}_i))^2 + (cos(v_i) - cos(\hat{v}_i))^2 \right) ,$$

The u–matrix visualization of the resulting map is shown in Figure 1 (left hand side).

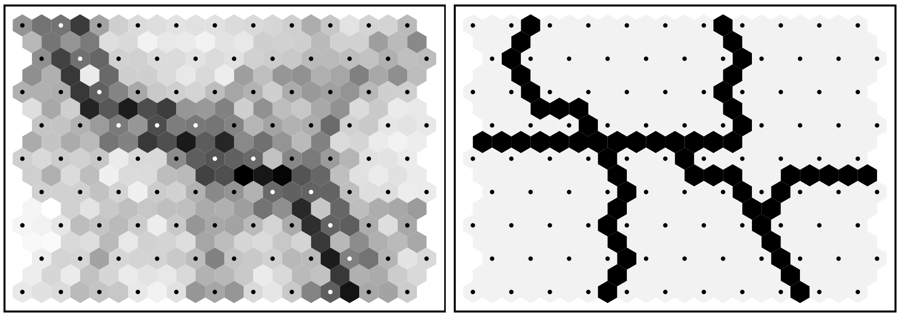

Figure 1: **Example 1: RNA molecules.** Left hand side: u–matrix visualization of the SOM for the molecular example. Right hand side: visualization of the cluster borders (see text).

After computing the stochastic matrix, an implementation of the dynamical approach generates the following eigenvalues (the five large gaps are marked in the table):

k	1	2	3	4	5	6	7	8	9	10	...
λ_k	1.00	0.90	0.82	0.67	0.65	0.54	0.45	0.44	0.38	0.37	...

Beyond eigenvalue number 6, there are only small gaps. Figure 2 shows a plot of all eigenvalues.

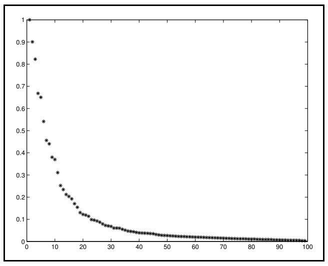

Figure 2: **Example 1: RNA molecules.** Plot of all eigenvalues

As a result, the algorithm generates four possible clusterings, with at least two clusters. If we look at the last large gap between eigenvalue number 6 and number 7, we get the clustering that is shown in Figure 1 (right hand side).

[5] Several tests have shown, that our algorithm is quite robust: If we use grid sizes that differs no more than 30-40%, we get the same clustering of the input space.

Example 2: Health Insurance In the next example, we have a clustering of customers of an German insurance company. Each of the 32000 customers used for the computation of the map, is described by 181 attributes (e.g. age, sex, occupation). After suitable transformations, all attributes are ordinal or metric and normalized.

Again, we use a 11×9 grid and a log-inverse radius function. For the distance function, we use the euclidean distance. The u–matrix visualization of the resulting map is shown in Figure 3 (left hand side).

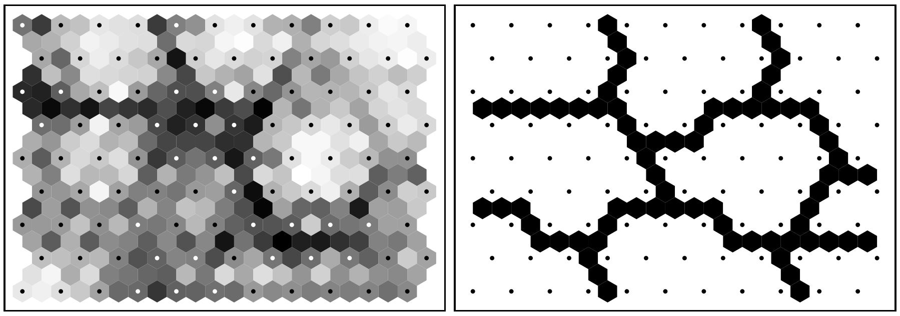

Figure 3: **Example 2: Health Insurance.** Left hand side: u–matrix visualization of the SOM for the insurance example. Right hand side: visualization of the cluster borders (see text).

We compute the stochastic matrix and use again our dynamical algorithm, getting the following eigenvalues (the two large gaps are marked in the table):

k	1	2	3	4	5	6	7	8	9	10	11	...
λ_k	1.00	0.94	0.89	0.83	0.74	0.69	0.67	0.61	0.55	0.46	0.42	...

Beyond eigenvalue number 10, there are only small gaps. Figure 4 shows a plot of all eigenvalues.

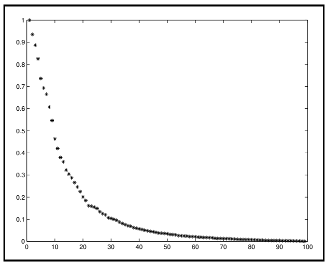

Figure 4: **Example 2: Health Insurance.** Plot of all eigenvalues

If we look at the last large gap between eigenvalue number 9 and number 10, we get the clustering that is shown in Figure 3 (left hand side).

For the sake of comparison, we want to report that the same clustering has been found in the past by using a non–automated, visualization based method. That method, however, has required one week of interactive work.

Summarizing, in both applications the obtained clusterings are not only reasonable, if one looks at the u–matrix visualization, but can also be justified by expertknowledge and statistical classification methods, e.g. a discriminant analysis [2]. The time needed to compute the stochastic matrix and to identify the clusters by our dynamical algorithm is negligible in comparison with the computing time spent for the SOM.

6 Conclusion

The paper presents a powerful, generally applicable approach to automatic cluster identification in the setting of Self-Organizing Maps. The described first algorithm already speeds up the cluster identification process considerably. Thus, the door has been opened for further applications of SOMs in the field of Data Mining. Future work will focus on further improvements of the suggested algorithm and its use in extended fields of application.

Acknowledgments. The first author (T.G.) was partially supported by RISK–CONSULTING, PROF. DR. WEYER, KÖLN. The second author (W.H.) was supported within the DFG–Schwerpunkt "Ergodentheorie, Analysis und effiziente Simulation dynamischer Systeme" under Grant De 293.

References

[1] A.Ultsch and D.Korus. Integration of neural networks with knowledge-based systems. In *Proc. IEEE Int.Conf.Neural Networks, Perth*, 1995.

[2] B.D.Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[3] B.S.Duran and P.L.Odell. *Cluster Analysis.* Springer, Berlin, 1974.

[4] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in nearly uncoupled Markov chains. Accepted in Lin. Alg. Appl., Available via http://www.zib.de/bib/pub/pw, 1998.

[5] G.Deboeck and T.Kohonen (Eds.). *Visual Explorations in Finance using Self-Organizing-Maps.* Springer, London, 1998.

[6] Wilhelm Huisinga, Christoph Best, Rainer Roitzsch, Christof Schütte, and Frank Cordes. From simulation data to conformational ensembles: Structure and dynamic based methods. To appear in J. Comp. Chem. 1999. Available via http://www.zib.de/bib/pub/pw, 1998.

[7] J.M.Buhmann. Learning and data clustering. Technical report, University of Bonn, 1994.

[8] J.Vesanto. Som-based data visualization methods. *Intelligent Data Analysis*, (3):111–126, 1999.

[9] S.Kaski. *Data Exploration Using Self-Organizing Maps.* PhD thesis, Helsinki University of Technology, 1997.

[10] T.Kohonen. *Self-Organizing Maps.* Springer, Berlin, 2nd edition, 1997.

[11] U.M.Fayyad, G.Patetsky-Shapiro, P.Smyth, and R.Uthurusamy (Eds.). *Advances in Knowledge Discovery and data Mining.* AAAI Press / The MIT Press, California, 1996.