Analysis of Complex Systems using the Self-Organizing Map

Esa Alhoniemi, Olli Simula and Juha Vesanto
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 2200, FIN-02015 HUT, Finland
e-mail {Esa.Alhoniemi, Olli.Simula, Juha.Vesanto}@hut.fi

3rd July 1997

Abstract

The Self-Organizing Map (SOM) is a powerful neural network method for the analysis and visualization of high-dimensional data. It maps nonlinear statistical relationships between high-dimensional input data into simple geometric relationships on a usually two-dimensional grid. The mapping roughly preserves the most important topological and metric relationships of the original data elements and, thus, inherently clusters the data. The need for efficient data visualization and clustering is often faced, for instance, in the analysis of various engineering problems. In this paper, the use of the SOM based methods in analysis, monitoring and modeling of complex industrial processes is discussed.

1 Introduction

In modeling and control of complex systems, it is usually assumed that a global, analytical system model can be defined. However, many industrial processes are so complicated that a global model cannot be built. In such cases, Artificial Neural Networks (ANNs) can successfully be used. ANNs build models directly based on process measurements, and thus provide a means to analyze processes without explicit physical process model. ANNs can also be used as "soft sensors" to estimate signal values or process variables that can only be measured indirectly or off-line. The use of the ANNs, however, requires that a large amount of good quality, stable, numerical data describing the process are available.

The Self-Organizing Map (SOM) [8] is a neural network algorithm which is based on unsupervised learning in a data-driven way. Unlike supervised learning methods, the SOM can be used for clustering data without knowing the class memberships of the input data. It can, thus, be used to detect features inherent to the problem. The SOM has been successfully applied in various engineering applications [9] covering, for instance, areas like pattern recognition, image analysis, process monitoring and control, and fault diagnosis [14, 15, 16]. The SOM has also proven to be a valuable tool in data mining and knowledge discovery with applications in full-text and financial data analysis [4, 9].

In this paper, SOM based methods in the analysis of complex systems are discussed. Special emphasis is on industrial applications in which a lot of measured information is available from automation systems.

2 The Self-Organizing Map as an analysis and visualization tool

The SOM consists of a regular, usually 2-dimensional, grid of neurons. Each neuron i of the SOM is represented by an n-dimensional weight, or model vector, $\mathbf{m}_i = [m_{i1}, \ldots, m_{in}]^T$ where n is equal to the dimension of the input vectors. The weight vectors of the SOM form a codebook. The SOM algorithm performs a topology preserving mapping from the high-dimensional input space onto map units so that relative distances between data points are preserved. Data points lying near each other in the input space will be mapped onto nearby map units. The SOM can thus serve as a clustering tool of high-dimensional data. It also has capability to generalize, i.e. the network can interpolate between previously encountered inputs.

The neurons of the map are connected to adjacent neurons by a neighborhood relation, which dictates the topology of the map. Usually rectangular or hexagonal topology is used. Immediate neighbors (adjacent neurons) belong to the neighborhood N_i of the neuron i. In the basic SOM algorithm, the topological relations and the number of neurons are fixed from the beginning. The number of neurons may vary from a few dozens up to several thousands. The number of neurons determines the granularity of the mapping, which affects accuracy and generalization capability of the SOM.

An example of applying the SOM in industrial process analysis is shown in Figure 1. The different stages in the Figure are: (1) Data processing (acquisition, preprocessing, feature extraction), (2) map training, (3) validation, and (4) visualization.

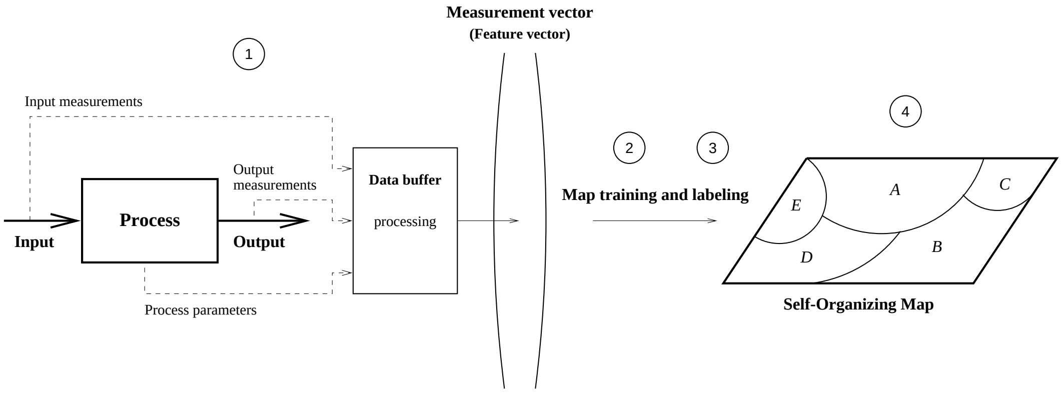

Figure 1: Stages of the application of the SOM in industrial process analysis.

During iterative training procedure, the SOM forms an elastic net that folds onto the "cloud" formed by input data. The net tends to approximate the probability density of the data [8]: the codebook vectors tend to drift there where the data are dense, while there are only a few codebook vectors where data are sparse. At each training step, one sample vector $\mathbf{x}$ is randomly drawn from the input data set. Distances (i.e., similarities) between the $\mathbf{x}$ and all the codebook vectors are computed. The best-matching unit (BMU) is the map unit whose weight vector is closest to the $\mathbf{x}$. After finding the BMU, the weight vectors of the SOM are updated. The BMU and its topological neighbors are moved closer to the input vector in the input space.

The quality of the mapping is usually determined based on (1) precision and (2) topology preservation. The former can be measured using average quantization error: the average distance between the input vectors of the testing set and the corresponding BMUs. Different topology measures have been studied e.g. by Kiviluoto [7] and Kaski and Lagus [5], who proposed a goodness meter which combined both properties.

The SOM can be interpreted by naming the units according to input vectors, whose type (e.g., class) is known. For each such vector, the BMU is determined and named after the vector. The labeling gives physical interpretation of the network. If labeled vectors are not available, the map can be interpreted by direct inspection of the weight vectors and clusters on the map. This is easiest to accomplish using different visualization techniques detailed in the next section. Also automatic interpretation of the map is possible using fuzzy rules as done by Pedrycz and Card [10].

The SOM can be used efficiently in data visualization due to its ability to approximate the probability density of input data and to represent it in two dimensions. In the following, several ways to visualize the network are introduced using a simple application example, where a computer system in a network environment was measured in terms of utilization rates of the central processing unit and traffic volumes in the network. The SOM was used to form a representation of the characteristic states of the system.

The unified distance matrix (u-matrix) method by Ultsch [17] visualizes the structure of the SOM. The u-matrix of the example system is shown in Figure 2a. The lighter the color between two map units is, the smaller is the relative distance between them. Uniform areas can clearly be detected on the map. These areas can also be labelled, as shown in the figure, based on the interpretatation of the input data.

Operating point and trajectory can be used to study the behavior of a process in time. The operating point of the process is the BMU of the current measurement vector. The location of the point corresponding to the current process state can be visualized on the SOM. A sequence of operating points in time forms a trajectory on the map depicting the movement of the operating point. A trajectory of the example system is plotted on top of u-matrix in Figure 2a.

Component plane representation visualizes relative component values of the weight vectors of the SOM. It is a "sliced" version of the SOM, where each plane shows the distribution of one weight vector component. Using the component plane representation dependencies between different process parameters can be investigated. The component planes of the example system are presented in Figure 2b. The lighter the color is, the smaller is the relative component value.

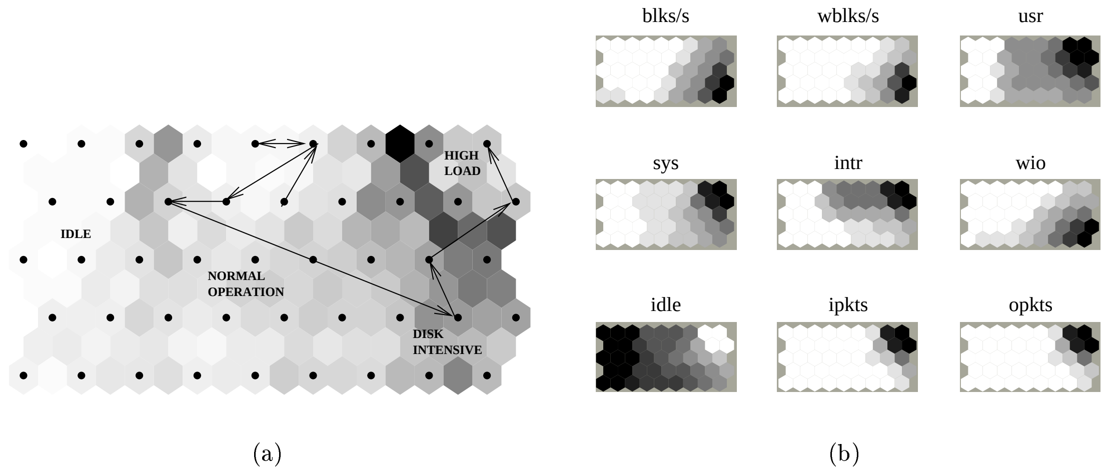

Figure 2: Different visualizations of the SOM. Trajectory on top of labeled u-matrix (a) and planes representation (b). In Figure (a) the black spots denote map units.

3 Applying the SOM to industrial systems

Depending on the system to be analysed, input and output measurements as well as process parameters can be used as input data for the SOM, as shown in Figure 1.

In process monitoring, two different approaches can be distinguished: (1) on- or off-line process analysis and (2) fault detection and possible identification in process behavior. Representative examples and many references can be found in articles by Kohonen *et al.* [9] and Simula and Kangas [15].

3.1 Process analysis using the SOM

In chemical process industry, for instance, the SOM can be applied to on-line observation of processes as well as off-line analysis of process data. Because clear faults seldom occur in such processes and are typically quite uninteresting, faulty situations should be filtered out from the training data set to be able to analyze normal operation more accurately.

In on-line use, the SOM is used to form a display of the operational states of the process. The current process state and its history in time can be visualized as a trajectory on the map. This allows efficient tracking of the process dynamics. The SOM facilitates understanding of processes so that several variables and their interactions may be inspected simultaneously. In off-line analysis, the SOM is also a highly visual data exploration tool. Nonlinear dependencies between process variables can be effectively presented using the component planes of the map.

Kasslin *et al.* [6] used the SOM to monitor the state of a power transformer and to indicate when the process was entering a non-desired state represented by a "forbidden" area on the map. Tryba and Goser [16] applied the SOM in monitoring of a distillation process and discussed its use in chemical process control in general.

In *fault detection* , the SOM can be trained using measurement vectors describing normal operation only. Thus, the SOM is trained to form a mapping of the "normal operation" input space. A faulty situation can be detected by monitoring the quantization error (distance between input vector and BMU). Large error indicates that the process is out of the normal operation space. For example, Alander *et al.* [1] and Harris [2] have used the SOM for this purpose.

The problem of *fault detection and identification* is more difficult. The SOM should be trained using all possible data describing both normal and abnormal situations of the process. If faulty situations are rare, measurements describing simulated faults may be added. Map units representing faulty states of the process may be labeled according to known samples. The monitoring is based on tracking of the operation point: location of the point on the map indicates the process state and the possible fault type. Vapola *et al.* [18] constructed a two-level SOM model, which was used first to detect and then to identify fault conditions in an anesthesia system.

3.2 Process modeling using the SOM

A general regression of y on $\mathbf{x}$ is usually defined as $\hat{y} = E(y|\mathbf{x})$. That is, the expectation of the output y given the input vector $\mathbf{x}$. To motivate the use of SOM for regression, it is worth noting that the codebook vectors represent local averages of the training data.

The SOM can be used for predicting, for example, the output quality of a process given the measurements of incoming raw material characteristics and process parameter settings [3]. Regression is accomplished by searching for the BMU using the known vector components of $\mathbf{x}$. As an output, an approximation of the unknown components of the codebook vector are given (see Figure 3).

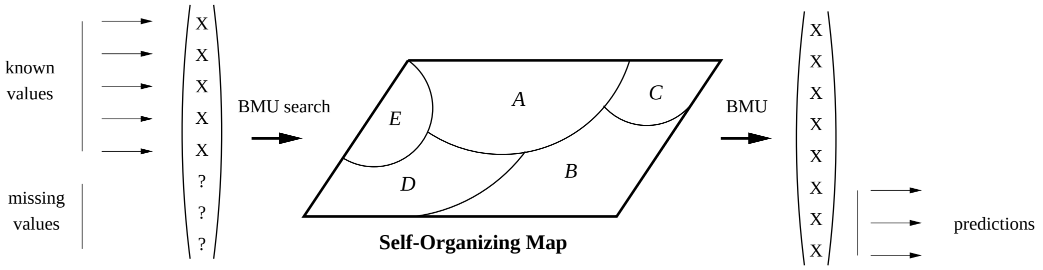

Figure 3: Prediction of missing components of the input vector.

The accuracy of the SOM model can be increased by building local models for the data in the

Voronoi sets of the SOM. The Voronoi set V_i of map unit i is a set of vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, for which the codebook vector $\mathbf{m}_i$ is closest.

The Voronoi sets provide a partitioning of the input data into disjoint sets. Each set contains points that are near each other in the data space. Subsets are modeled by independent local models, which together are considered a solution to the modeling problem. Each model is based on local data set $\mathbf{V}_i$ only. This kind of approach could be called divide-and-conquer modeling. The models are not constrained to be of any specific form, or not even of similar form. In our experiments, only simple local linear models have been considered.

A total least squares type of linear regression can be performed using Principal Component Analysis (PCA) in model fitting. This approach allows measurement errors also in inputs while the usual least squares approach assumes that the input variables are accurate and there is error in the output variables only [12]. Combining these two modeling methods takes advantage of the nonlinear elasticity of the SOM as well as the local efficiency of the PCA. Also, the topology preservation property of the SOM projection can be incorporated by allowing neighboring data sets and models to interact in some way. Ritter *et al.* [13] used this approach in local modeling of three-dimensional working space of a robot arm.

It is often desirable to know the behavior of a system under small changes made in the system parameters. This is especially the case in industrial environments, where noise is present both in measurements and in operating conditions. In process control, the state of the process is desired to be moved to such a direction that better quality is achieved. The operation point needs to be stable: small random fluctuations in input parameters must not cause large changes in output parameters.

The model described above can be used to investigate leverage effects of small changes made in one of the process parameters. This is possible because the system cannot reach all the possible values in the space defined by the measurements, but may be limited to a low-dimensional manifold. The state space, or the space of possible values, is constrained by the characteristic behavior of the system. As a small change in one of the measurements is imposed, the BMU changes to another map unit. By tracking the change of the BMU caused by the change of the parameters, the mutual nonlinear dependence of the parameters is revealed; one is "surfing" on a low-dimensional manifold defined by the SOM projection [3].

4 Conclusions

In this paper, the use of the SOM in analysis, monitoring and modeling of various industrial applications has been presented. The SOM is especially suitable in tasks which require processing of large amounts of numerical data. The method is readily explainable, simple and highly visual.

The SOM provides data-driven approach to process monitoring. When using the SOM, it is not necessary to define process model analytically. The SOM has the desirable feature of describing nonlinear relationships between large number of parameters and variables of complex systems phenomenologically. By using a history of measurements, dynamical behavior of the process can be introduced into the map, or set of maps. This approach has been used to model the sequence of states and based on that to predict the future state in the system operation [11, 14].

The SOM facilitates visual understanding of processes. For instance, process operation personnel may learn to adjust the control variables in such a way that the operation point, or trajectory, stays in the desired region on the map. In this way, correct control action may be easily learned based on the visual output.

The construction of feature vectors allows fusion of different data and measurement sources. Information from various databases can be integrated, e.g. using SOMs in a hierarchical way. For instance, in the analysis of complex industrial processes, technical, economical, and environmental data can be combined [19]. This allows the analysis and simulation of various effects in the entire field of industry, e.g., the influence of various technical investments can be analyzed in the factory level.

References

[1] J. T. Alander, M. Frisk, L. Holmström, A. Hämäläinen, and J. Tuominen. Process error detection using self-organizing feature maps. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, volume II, pages 1229–1232, Amsterdam, Netherlands, 1991. North-Holland.

[2] T. Harris. A Kohonen S.O.M. based, machine health monitoring system which enables diagnosis of faults not seen in the training set. In *Proc. IJCNN-93-Nagoya, Int. Joint Conf. on Neural Networks*, volume I, pages 947–950. IEEE Service Center, 1993.

[3] J. Hollmén and O. Simula. Prediction models and sensitivity analysis of industrial production process parameters by using the self-organizing map. In *1996 IEEE Nordic Signal Processing Symposium Proceedings (NORSIG'96)*, pages 79–82, 1996.

[4] S. Kaski. Data exploration using self-organizing maps. *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82*, March 1997. DTech Thesis, Helsinki University of Technology, Finland.

[5] S. Kaski and K. Lagus. Comparing self-organizing maps. In *Proceedings of ICANN'96*, pages 809 – 814, 1996.

[6] M. Kasslin, J. Kangas, and O. Simula. Process state monitoring using self-organizing maps. In *Artificial Neural Networks, 2*, volume II, pages 1531–1534, Amsterdam, Netherlands, 1992. North-Holland.

[7] K. Kiviluoto. Topology preservation in self-organizing maps. Technical Report A29, Helsinki University of Technology, Laboratory of Computer and Information Science, 1995.

[8] T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, Heidelberg, 1995.

[9] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358 – 1384, October 1996.

[10] W. Pedrycz and H. C. Card. Linguistic interpretation of self-organizing maps. In *Proc. of International Conference on Fuzzy Systems '92*, 1992.

[11] J. C. Principe and L. Wang. Non-linear time series modeling with Self-Organization Feature Maps. In *Proc. NNSP'95, IEEE Workshop on Neural Networks for Signal Processing*, pages 11–20. IEEE Service Center, September 1995.

[12] C. R. Rao and H. Toutenburg. *Linear models: least squares and alternatives.* Springer-Verlag, New York Inc., 1995.

[13] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps.* Addison-Wesley Publishing Company, 1992.

[14] O. Simula, E. Alhoniemi, J. Hollmén, and J. Vesanto. Monitoring and modeling of complex processes using hierarchical self-organizing maps. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'96)*, volume Supplement, pages 73–76, May 1996.

[15] O. Simula and J. Kangas. *Neural Networks for Chemical Engineers*, volume 6 of *Computer-Aided Chemical Engineering*, chapter 14, Process monitoring and visualization using self-organizing maps. Elsevier, Amsterdam, 1995.

[16] V. Tryba and K. Goser. Self-Organizing Feature Maps for process control in chemistry. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 847–852, Amsterdam, Netherlands, 1991. North-Holland.

[17] A. Ultsch and H. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proc. INNC'90, Int. Neural Network Conf.*, pages 305–308, Dordrecht, Netherlands, 1990. Kluwer.

[18] M. Vapola, O. Simula, T. Kohonen, and P. Meriläinen. Representation and identification of fault conditions of an anaesthesia system by means of the Self-Organizing Map. In M. Marinaro and P. G. Morasso, editors, *Proc. ICANN'94, Int. Conf. on Artificial Neural Networks*, volume I, pages 350–353, London, UK, 1994. Springer.

[19] J. Vesanto, P. Vasara, R.-R. Helminen, and O. Simula. Integrating environmental, technologigal and financial data in forest industry analysis. In *Proceedings of SNN'97 Europe's Best Neural Networks Practice*, Amsterdam, The Netherlands, May 1997.