The Neuro-Data-Mine

A. Ultsch

Philipps-University of Marburg, Department of Computer Science
Hans-Meerwein-Str., 35032 Marburg
Fax: +496421/2828902 , E-mail: ultsch@mathematik.uni-marburg.de

Abstract: In this paper we present the Neuro-Data-Mine (NDM) an implementation of a model for Data Mining using neural networks. The model describes a stepwise transformation of raw data into symbolic descriptions of these data. The NDM performs this transformation by a combination of subsymbolic and symbolic information processing. It combines an Emergent Feature Map with Machine Learning algorithms to gain hypotheses about the data. This method combines the advantages of both symbolic and subsymbolic working, while avoiding most of their disadvantages. We consider the connection of these two levels of information as the key to the discovery of knowledge in data. The NDM has been successfully applied to a number of problems.

1 Introduction

In many cases data analysis is done by classical statistics. In general the statistical process starts with a hypothesis, then collects the necessary data and at last tries to falsify the hypothesis using statistical tests. Nowadays in many fields of science as well as in business we are confronted with growing amounts of data without having any prior hypotheses about them. In this situation the statistical method has two major disadvantages, first the collection and preparation of the data for the tests is a very time consuming task, and second, finding meaningful hypotheses about the data is not a trivial task. The first problem is more a technical one and can be solved by means of OLAP- (Online analytical processing) and Data Warehouse- (DW) systems. The second problem is less easy to overcome, because OLAP and DW can only help to navigate faster and with greater comfort through the data, besides some aggregations of the data they provide no new information, especially nothing we could call knowledge.

The key problem is to find meaningful hypotheses about structures that may be in the data. These hypotheses are for example symbolic descriptions of certain aspects of the probability density function underlying the data. Because the data is subsymbolic a transformation from subsymbolic to symbolic is necessary (see chapter 3.4). Self-Organizing Feature Maps [Kohonen 82] are very helpful in this respect. If appropriately used, they exhibit the ability of emergence [Ultsch 99]. I. e. with the cooperation of many neurons Emergent Feature Maps are able to build structures on a new higher level. The U-Matrix-method [Ultsch 90] visualizes these structures of the high-dimensional input-space that otherwise would be invisible. A Knowledge Conversion algorithm transforms the structures seen into a symbolic description of the relevant properties of the dataset. Such systems which provide this transformation of subsymbolic information into symbolic information we call hybrid systems (see chapter 3.1).

In chapter two we shortly introduce our approach to Data Mining and Knowledge Discovery and present a model for Data Mining (chapter 2.1). Chapter 2.2 describes the NDM-system, a possible implementation of this model, which manifests the subsymbolic part of the model by using a Emergent Feature Map, because of their ability to adapt to the structure of a given set of data and represent this on a emergent level. Chapter three discusses the basic principles and used methods of the NDM-system. Chapter four lists some applications and chapter five contains the discussion.

2 Data Mining

Since the use of the term Data Mining is quite divers, we give a short definition in order to specify our approach to Data Mining and Knowledge Discovery. A more detailed description can be found in [Ultsch 99a]. We define Data Mining as the inspection of a large dataset with the aim of Knowledge Discovery. Knowledge Discovery is the discovery of new knowledge, i. e. knowledge that is unknown in this form so far. This knowledge has to be represented symbolically and should be understandable for human beings as well as it should be useable in knowledge-based systems.

Central issue of Data Mining is the transition from data to knowledge. Symbolically represented knowledge - as sought by Data Mining - is a representation of facts in a formal language such that an interpreter with competence to process symbols can utilize this knowledge [Ultsch 87]. I. e. in particular, human beings must be able to read, understand and evaluate this knowledge. The knowledge should also be useable by knowledge-based systems. The knowledge should be useful for analysis, diagnosis, simulation and/or prognosis of the process which generated the dataset. We call the transition from data , respectively an unfit knowledge representation, to useful symbolic knowledge Knowledge Conversion [Ultsch 98].

2.1 Steps of Data Mining

Our model of Data Mining can be described as process, which gradually extracts relevant descriptions from the raw data. The model we use roughly consists out of two parts, a subsymbolic part, where subsymbolic descriptions of the data are generated by means of Artificial Neural Nets, and a symbolic part, where symbolic descriptions are generated out of the subsymbolic clustering.

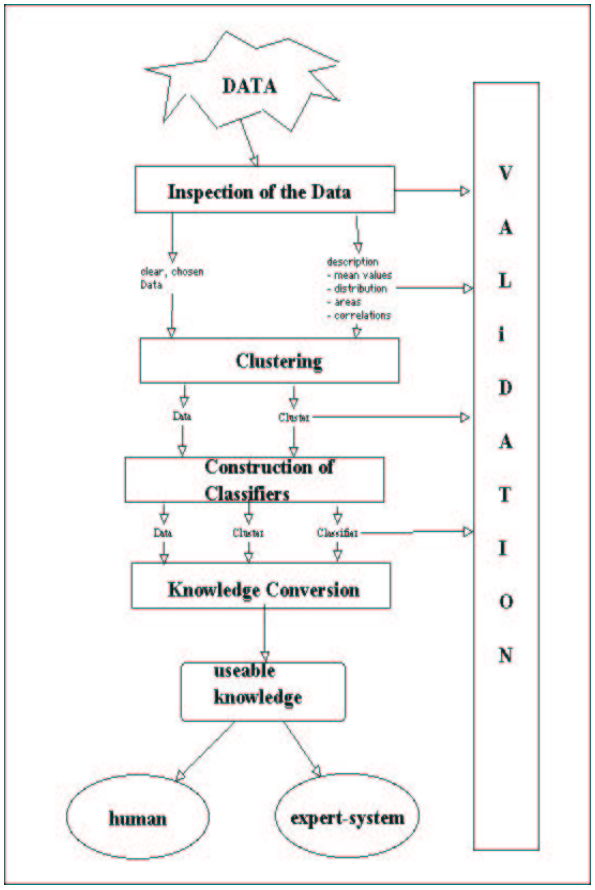


Figure 1: The Data Mining-model

An overview of this Data-Mining-model shows the following five main transformation steps (see figure 1):

1. Inspection of the dataset
2. Clustering
3. Construction of classifiers
4. Knowledge Conversion
5. Validation

Unfortunately it has to be stated that in many commercial Data Mining tools there is no Knowledge Conversion [Gaul 98]. The terms Data Mining and Knowledge Discovery are often used in those systems in an inflationary way for statistical tools enhanced with a fancy visualization interface [Woods, Kyral 98]. The difference between exploratory statistical analysis and Data Mining lies in the aim which is sought. Data Mining aims at Knowledge Discovery.

2.2 The NDM-system

The modules of the NDM reflect directly the transformation steps of the NDM-model (see figure 2):

1. Statistics:	Statistical preprocessing of the input data
2. U-Matrix:	Generation of the U-Matrix
3. VisClass:	Visualisation and Classification of the U-Matrix
4. Sig*:	Generation of meaningful symbolic descriptions of the detected classes
5. Rule Analysis:	Measuring the quality of the classification

The subsymbolic processing section consists of a Self-Organizing Map (SOM), which is fed by the preprocessed data. This preprocessing can also be done by using conventional statistics or by using Artificial Neural Nets [Ultsch Halmans 91a]. The structure of the trained SOM is then visualized by means of the U-Matrix method for the purpose of clustering. The result of the subsymbolic processing section is a subsymbolic classificator for the data.

The symbolic section of the NDM begins after the clustering is accomplished. Then meaningful symbolic descriptions of the clusters of the subsymbolic data can be extracted. This is done by the inductive Machine Learning algorithm called **SIG*** [Ultsch 91a]. It takes the training data with the classification detected through the learned SOM as input and generates rules for characterizing and differentiating the classes of the data (See chapter 3.4.2). The result is a symbolic classificator for the given data.

Finally the NDM allows to validate the quality of the symbolic classificator generated in this way in terms of sensitivity, specificity and accuracy.

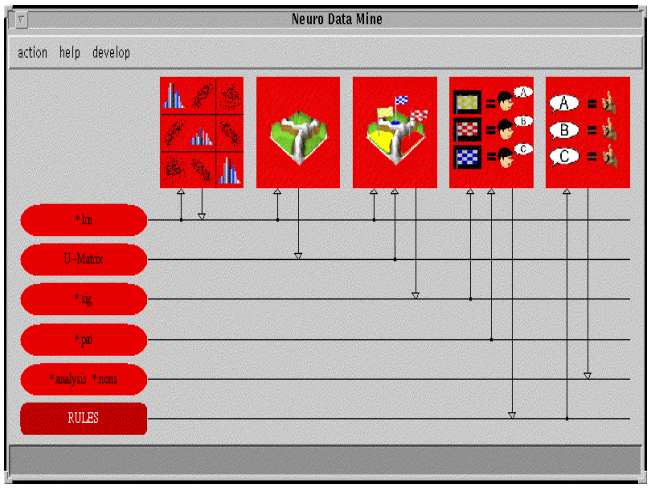


Figure 2: The modules of the NDM

3 Basic Principles

3.1 Hybrid architecture

The notion „hybrid“ stands here for an amalgamation or crossing of different kinds of technologies: On the one hand a special Connectionist Model and on the other hand “classical” Artificial Intelligence (AI). The aim of this combination is the assumption, that both technologies are different but complementary [Goonatilake Khebbal 95]. AI technologies are said to be easily interpreted, easily controlled and contain a high level of knowledge abstraction. Connectionist Models, in contrast, show advantages with regard to learning capabilities and robustness and furthermore, they claim to be more error tolerant than AI technologies [Gallant 93].

Beside the above mentioned pragmatical differences a principal, i.e. paradigmatical, difference between Connectionist Models and AI technologies is claimed [Newell 80, Smolensky 88]. One of these differences should concern the computational power, i.e. what is possible to be computed by computer systems in principal. Furthermore, it is assumed that by renunciating to understand the derivation of the solution, a broader class of problems can be solved. The difference mainly should lie in the difference between subsymbolic knowledge representation in Connectionist Models vs. the symbolic knowledge representation in AI technologies [Honavar Uhr 95].

Regarding the integration of Connectionist Models with symbolic AI techniques two different approaches can be distinguished: cooperative and true hybrids. Cooperate means that a system works with symbolic and subsymbolic representations of knowledge, maybe in different modules, but no transition between subsymbolic and symbolic knowledge representation takes place. Opposed to that true Hybrid systems are those systems that manifest the conversion between subsymbolic and symbolic knowledge.

3.2 The Self-Organizing Map (SOM)

Self-Organizing Feature Maps were developed by Teuvo Kohonen in 1982 and should, to our understanding, exhibit the following interesting and non-trivial property [Kohonen 82]: the ability of emergence through self-organization. Self-organization means the ability of a biological or technical system to adapt its internal structure to structures sensed in the input of the system. For further reference on the emergence of SOM see [Ultsch 99].

Kohonen‘s Self-Organizing Feature Maps belong to the class of unsupervised learning ANNs. For an overview on ANN types see for example [Ultsch 91, Ritter et al 90, Lipp 87]. The SOM consist basically of two layers of so called units or neurons. The input layer consists of N neurons corresponding to the real-valued input vector of dimension N. These units are connected to a second layer of neurons U. By means of lateral connections, the neurons in U form a lattice structure of dimensionality M. Typically M is much smaller than N.

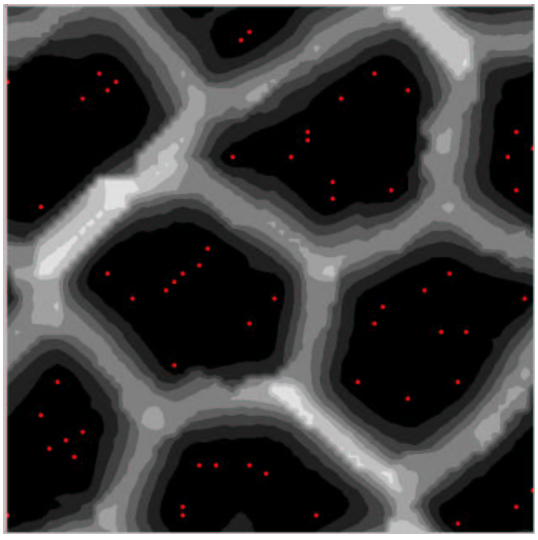
When a input data vector is presented to the network, it responds with the location of a unit in U, that corresponds most closely to the presented input. This is called the bestmatching neuron or for short the bestmatch. As a result of the leaning process, i.e. the presentation of all input vectors and the adaptation of the weight vectors, the SOM generates a mapping from the input space $\Re^n$ onto the lattice U with the property, that the topological relationships in input space are preserved in U as good as possible [Ritter Schulten 86, Kohonen 89]. Similar input data should correspond to bestmatches in U that are close together on the lattice. The main problem of the SOM is that the structures in the data organized by the SOM can not be seen, i.e. *no clustering can be detected*.

3.3 The U-Matrix methods

In order to detect and display the structures a set of methods, called U-Matrix methods (UMMs) has been developed 1990 by Ultsch [Ultsch 90]. The idea of UMMs is to visualize the Feature Map's topology. Analyzing the weights at each point of the grid with respect to their neighbors and then displaying the distance between two neighbor units as height, leads to an interpretable, 3-dimensional landscape of the Kohonen Map (for detail see for example [Ultsch 92]). This diagram we call *Unified distance matrix* or short U-Matrix. The U-Matrix contains a geometrical approximation of the vector distribution in the unit layer U. This display has valleys where the vectors in the lattice are close to each other and hills or walls where there are larger distances, indicating dissimilarities in the input data.

To explore the properties of a n-dimensional data space we propose to interpret the landscape of the U-Matrix. Given a n-dimensional data set, the question is if there exists any structure in form of subsets i.e. clusters of data that are very similar to each other. The U-Matrix should be interpreted as follows: If a subset of input data falls into a valley on the U-Matrix surrounded by walls then this indicates a cluster containing similar vectors. Neighbors on the U-Matrix are close to each other in the $\Re^n$. The dissimilarity among the different clusters is indicated by the height of the walls or hills on the U-Matrix. In order to cluster a given data set it should be noted in particular, that no previous information on the number of clusters is needed.

U-matrices give a picture of the topology of the unit-layer and therefore also of the topology of the input space. Altitude in the U-Matrix encodes dissimilarity in the input space. Valleys in the U-Matrix (i.e. low altitudes) correspond to input-vectors that are quite similar.



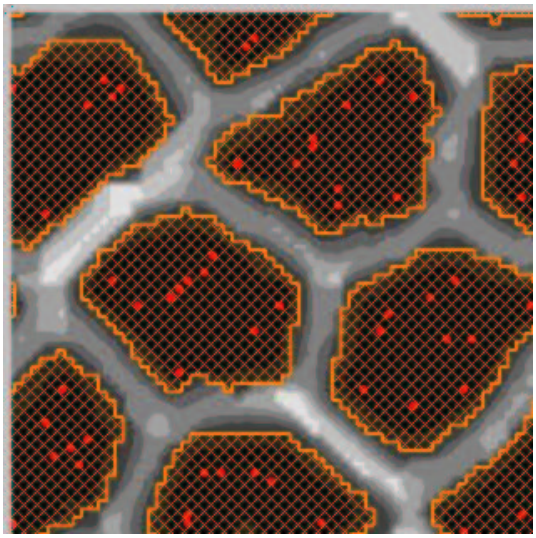


Figure 3a: U-Matrix , **3b:** Visual clustering using SOM with a U-Matrix-method

3.4 Symbolic vs. Subsymbolic information

The word "symbol" as defined in a dictionary is an "object or a process that serves as a place-holder for a psychic entity which cannot be perceived directly" [Duden 95]. In computer science „object or process" is usually defined as an element of an alphabet i.e. a sign. This sign, for example a letter, is regarded atomic and cannot be divided into subelements without loss of the meaning of the sign. As „psychic entity which cannot be perceived directly" the semantics of the sign or words build out of signs can be regarded.

A subsymbolic representation gives rise to some expectations in Connectionist Models. A subsymbolic representation is presumably error tolerant since the modification of one microfeature will in general have no sensible consequences on the represented symbol. In symbolic AI techniques for the acquisition of "new" knowledge exist. These techniques are known for example as Machine Learning (ML) algorithms [Shavlik Dietterich 90]. Assuming a symbolic representation of knowledge one could object that by Machine Learning techniques only known symbols are agglomerated to form linear combinations of known entities. Something really new in the sense of new entities or the detection of new relationships is not to be expected from these technologies. The superposition of representations in the units implies for connectionistic systems that they can detect really new entities or new relations.

3.4.1 Extraction of symbolic information

As a first approach to generate rules from the classified data we used a well known Machine Learning algorithm: ID3 [QUINLAN 84, Ultsch Panda 91]. While being able to generate rules this algorithm has a serious problem [Ultsch 91]: it uses a minimization criterion that seems to be unnatural for a human expert. Rules are generated, that use only a minimal set of decisions to come to a conclusion. This is not what has to be done, for example, in a medical domain. Here the number of decisions is based on the type of the disease. In simple cases, i.e. where the symptoms are unanimous, very few tests are made, while in difficult cases a diagnosis must be based on all available information.

In order to solve this problem we have developed a rule generation algorithm called SIG*, that takes the significance of a symptom (range of a component value) into account [Ultsch 91]. One of the data sets we tested the algorithm with, was the diagnosis of iron deficiency. We had a test set of 242 patients with 11 clinical test values each. The rules generated with SIG* showed, first, a high degree of coincidence with expert's diagnosis rules and, second, exhibited knowledge not prior known to us while making sense to the experts [Ultsch 91].

3.4.2 Rule generation with SIG*

SIG* has been developed in the context of medical applications [Ultsch 91a]. In this domain other rule-generating algorithms such as ID3 [Quinlan 83], for example, fail to produce suitable rules. **SIG*** takes a data set in the space $\Re^n$ that has been classified by SOM/UMM as input and produces descriptions of the classes in the form of decision rules. For each class an essential rule, called characterizing rule, is generated, which describes that class. Additional rules that distinguish between different classes are also generated. These are called differentiating rules. This models the typical differential-diagnosing approach of medical experts, but is a very common approach in other domains as well. The generated rules by **SIG***, in particular, take the significance of the different structural properties of the classes into account. If only a few properties account for most of the cases of a class, the rules are kept very simple.

Two central problems are addressed by the **SIG*** algorithm:

1. *how to decide which attributes of the data are significant to characterize each class,*
2. *how to formulate apt conditions for each selected significant attribute.*

In order to solve the first problem, each attribute of a class is associated with a "significance value". The significance value can be obtained, for example, by means of statistical measures. We assume a data set of case-vectors with attributes $Attr_{i.}$ and let SOM/UMM distinguish the classes Cl_k. Let SV_{ik} denote the significance value of $Attr_i$ in class C_k . In the matrix $SM=(SV_{ij})^{ixk}$ we call "significance matrix", the largest value in each row is marked, the significance values of the attributes are normalized and then ordered. As significant attributes for the description of a class, the attributes with the largest significance value in the ordered sequence are taken.

For the second problem we can make use of the distribution properties of the attributes of a class. A class is described by a number of conditions based on the attributes selected by the algorithm described above. The algorithm produces the essential description of a class. If the intersection of such descriptions of two classes is nonempty, a finer description of the borderline between the two overlapping classes is necessary. To the characterizing rule of each class a condition is added that is tested by a differentiating rule. A rule that differentiates between the classes A and B is generated by an analog algorithm as for the characterizing rules. As for significance values however, they may be measured between the particular classes A and B. The conditions are typically set stronger in the case of characterizing rules. To compensate this the conditions of the differentiating rules are connected by a logical ***OR***. The complete and formal description can be found in [Ultsch 91a].

3.4.2.1 Constructing Conditions for the Significant Attributes of a Class

A class is described by a number of conditions based on the attributes selected by the algorithm described above. If these conditions are too strong, many cases may not be correctly diagnosed. If the conditions are too soft, cases that do not belong to a certain class are erroneously subsumed under that class. The main problem is to estimate the distributions of the attributes of a class correctly. If no assumption on the distribution is made, the minimum and maximum of all those vectors that belong, according to SOM/UMM, to a

certain class may be taken as the limits of the attribute value. In this case a condition of the i-th attribute in the j-th class can look like:

$attribute_{ij}$ IN [min_{ij} , max_{ij}] .

But this kind of formalization of conditions likely results in an erroneous subsumption .

If a normal distribution is assumed for a certain attribute, we know from statistics that 95% of the attribute values are captured in the limits [$mean_{ij}$ -2**dev* ,$mean_{ij}$ +2**dev*], where *dev* is the value of the standard deviation of the attribute. For other assumptions about the distribution, two parameters *low* and *hi* may be given in **SIG*.** For this case the conditions generated are as follows:

$attribute_{ij}$ IN [$mean_{ij}$ + low * dev,
$mean_{ij}$ + hi * dev].

3.5 The rule analysis

The rules generated by SIG* can now be used as a symbolic classificator for the data. Because of that they can now be validated against the subsymbolic classificator generated before by means of the U-Matrix. The quality of the classificator is shown in terms of sensitivity and specificity. The overall quality of the classificator can be expressed by its accuracy and its predictive value.

- Sensitivity (for class j): $Sens_j = m_{jj} / m_j$
- Specificity (for class j):
$$Spez_j = \frac{\overline{m}_j - \overline{m}_{jj}}{\overline{m}_j} = \frac{(n - m_j) - (d_j - m_{jj})}{n - m_j}$$
- Predictive value (for class j): $\Pr v_j = m_j / d_j$
- Accuracy: $$Ak = \sum_{j=1}^{k} m_{jj} \Big/ n$$

Here n is the overall number of cases; m_j the number of cases belonging to class j ; m_{jj} the number of cases classified as j but belong to class j; d_j is the overall number of cases classified as class j (without any regard of their true class).

4 Applications

The NDM has been applied successfully to a variety of problem domains, for example the analysis of cerebrospinal fluid, the prediction of hail, biochemical analysis, marine geology and avalanche forecasting.

5 Conclusion

A key problem of Data Mining is the transformation between the subsymbolic level and the symbolic level. The results must be in a symbolic representation, for they are useable for humans or knowledge acquisition systems. In this paper we presented a model for Data Mining, which accomplishes this transformation from subsymbolic to symbolic and as an implementation of this model the NDM-system. We discussed the central issues of integrating successive transformation steps into a model for the whole Data Mining-process. Furthermore we discussed the integration of subsymbolic and symbolic information processing into a hybrid system and the clustering of the typically high-dimensional data by means of a Self-Organizing Map with a U-Matrix-method. This conversion between levels of representation we regard as crucial for Data Mining. For the found symbolic hypotheses we described the NDM module for the analysis, which calculates a measure of quality for these hypotheses.

6 References

[Duden 95] Duden, Bibliographisches Institut, Mannheim, 1995

[Gallant 93] Gallant S. I.: Neural Network Learning and Expert Systems, MIT Press, Cambridge, MA, 1993

[Gaul 98] Gaul, W. G. Classification and Positioning of Data Mining Tools, Herausforderungen der Informationsgesellschaft an Datenanalyse und Wissensverarbeitung 22. Jahrestagung Gesellschaft f. Klassifikation

[Goonatilake Khebbal 95] Goonatilake, S., Khebbal, S.: Intelligent Hybrid Systems: Issues, Classifications and Future Directions, in Goonatilake, S., Khebbal, S. (eds): Intelligent Hybrid Systems, Wiley and Sons 1995, pp 1 - 20.

[Honavar Uhr 95] Honavar, V., Uhr, L.:Integrating Symbol Processing Systems and Connectionist Networks, in Goonatilake, S., Khebbal, S.(eds): Intelligent Hybrid Systems, Wiley and Sons 1995, pp. 177 - 208.

[Kohonen 82] Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43-69, 1982.

[Kohonen 89] Kohonen, T.: Self-Organization and Associative Memory, Springer 1989

[Lipp 87] Lippmann, R.P.: An Introduction to Computing with Neural Nets, IEEE ASSP Magazine, April 1987, pp. 4-22.

[Newell 80] Newell A.: Physical symbol systems. Cognitive Science 4, 1980, pp. 135 -183.

[Quinlan 83] Quinlan, J.R. "Learning Efficient Classification Procedures and Their Application to Chess Endgames" in Machine Learning : An AI Approach, Vol. 1, Michalski, R. S., Carbonell, J.G., and Mitchell, T.M., eds., 1983.

[Quinlan 84] Quinlan J.R.: Learning Efficient Classification Procedures and their Application to Chess End Games in: Michalski R., Carbonell J.G., Mitchell T.M.: Machine Learning - An artificial intelligence approach,Springer Verlag, Berlin 1984

[Ritter et al. 90] Ritter, H., Martinez, T., Schulten, K.: Neuronale Netze. Addison Wesley

[Ritter Schulten 86] H.Ritter, K.Schulten: Kohonen's Self-Organizing Maps: Exploring their Computational Capabilities, Biol. Cybernetics Vol 54, 1986, pp.99-106.

[Smolensky 88] Smolensky, P.: On the proper treatment of connectionism. Behavioral and Brain Sciences 11, 1988, pp. 1-74.

[Ultsch 90] Ultsch A., Siemon H.P.: Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis, Proc. Intern. Neural Networks, Kluwer Academic Press, Paris, 1990, pp. 305 - 308

[Ultsch 91] Ultsch, A.: Konnektionistische Modelle und ihre Integration mit wissensbasierten Systemen, Habilitationsschrift, Univ. Dortmund, 1991.

[Ultsch 91a] Ultsch, A.: The Integration of Neuronal Networks with Expert Systems, Proc. Workshop on Industrial Applications of Neural Networks, Ascona, Vol III, September 1991 pp 3-7.

[Ultsch 92] Ultsch, A.: Self-Organizing Neural Networks for Visualization and Classification, Proc. Conf. Soc. for Information and Classification, Dortmund, April 1992.

[Ultsch 99a] Ultsch, A.:Data Mining und Knowledge Discovery mit Neuronalen Netzen, Technical Report, Department of Computer Science, University of Marburg, Hans-Meerwein- Str., 35032 Marburg

[Ultsch 99] Ultsch, A.: Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series, **in:** Kohonen Maps, Oja, E., Kaski, S. (Eds.), pp 33 - 46

[Ultsch Halmans 91a] Ultsch, A., Halmans, G.: Die Transformation experimenteller Verteilungen durch eine Self-Organizing Feature Map, in Ziegler, H.(Ed.): Konnektionismus: Beiträge aus Theorie und Praxis, Proc. ÖGAI, Wien 1991, pp. 32-40.

[Ultsch Panda 91] Ultsch, A., Panda, PG.: Die Kopplung konnektionistischer Modelle mit wissensbasierten Systemen, Tagungsband Expertenystemtage Dortmund, Februar 1991 VDI Verlag, pp 74-94.

[Woods Kyral 97] Woods, E., Kyral, E.: Data Mining, Ovum Evaluates, Catalumya Spain, 1997