Self-Organising Data Mining

F.Lemke, J.-A. Müller

This paper describes the possibility to widely automate the whole knowledge discovery process by applying self-organisation and other principles, and what is called self-organising data mining. There are different GMDH based modeling algorithms implemented - dimensionality reduction, missing value elimination, active neurons, enhanced network synthesis and creation of systems of equations, validation, combining of alternative models - to make knowledge extraction task appropriate, fast and easy-to-use even for large and complex systems.

1. Introduction

To derive knowledge from data is an interactive and iterative process of various subtasks and decisions and is called Knowledge Discovery from Data (KDD). The engine of Knowledge Discovery - where data is transformed into knowledge for decision making - is Data Mining. It is most important for a more sophisticated data mining to limit the user involvement in the entire data mining process to the inclusion of well-known a priori knowledge. Soft computing, i.e., Fuzzy Modelling, Neural Networks, Genetic Algorithms and other methods of automatic model generation, is a way to mine data by generating mathematical models from empirical data more or less automatically. The most recently developed tools for data analysis today known as data mining tools only begin the process of automating the search of models. Most tools have focused almost exclusively on building models. The most time-consuming process of data preparation, traditionally, is carried out by hand and is very hard to automate.
This paper describes the possibility to widely automate the whole knowledge discovery process by applying self-organisation and other principles, and what is called self-organising data mining.

2. Self-organising data mining (SODM)

In contrast to Neural Networks that use Genetic Algorithms as an external procedure to optimise the network architecture and several pruning techniques to counteract overtraining,
SODM introduces principles of evolution - inheritance, mutation and selection - for generating a network structure systematically enabling automatic model structure synthesis and model validation. Models are generated adaptively from data in form of networks of active neurons in an evolutionary fashion of repetitive generation of populations of competing models of growing complexity, their validation and selection until an optimal complex model - not too simple and not too complex - have been created. If this, but also data reduction, preprocessing and validation of model results is adjusted during the process of self-organisation, it is called self-organising data mining. Besides other steps, self-organising data mining algorithms realise in an objective way

- data transformation;
- preprocessing, such as elimination of missing values;
- data (dimensionality) reduction in state and/or sample space;
- choice of appropriate model and corresponding data mining algorithm;
- self-organisation of transfer functions (neurons);
- generation of alternative models with different variables and growing complexity in each layer;
- self-organisation of networks
- for multi-output systems: self-organisation of systems of networks (autonomous systems of equations);
- validation of generated models;
- self-organisation of hybrid models;
- using some control module on the base of generated models and/or predictions automatically to derive decisions.

3. Principles of self-organisation

The SODM approach is based on induction, which is composed of the cybernetic principle of self-organisation as an adaptive creation of a network without subjective points given:

First condition*: There is a very simple initial organisation (neuron) that enables the description of a large class of systems through its evolution*;
There are several linear or nonlinear reference functions. Components of input vector x_i, i=1, 2, ..., n, can be independent variables, functional terms or finite difference terms. For generation of systems of fuzzy rules the transfer function has to be $f(v_i,v_j)= \min (v_i,v_j)$ where v_i,v_j components of a fuzzy input vector.

Second condition*: There is an algorithm for the mutation of the initial or already evolved organisations of a population (network layer).*
Genetic Algorithms are working on more or less stochastic mutations of the model structure by means of crossover, stochastic changes of characteristic parameters and others. In the SODM approach, Ivakhnenko's GMDH principle is used, which has the following components:

1. to use in pairs the variables;
2. in a recursive way to solve a big system of Gaussian equations by many little systems of equations (3 to 7 equations), i.e. for every pair of variables gives the possibility to solve the interpolation task independent from the number of unknown parameters in an effective way;
3. because there is no practical way to check for every combination, the third part of GMDH is the multilayered approach: instead of generation of all possible combinations in one layer using the principal of selection of best models among all generated models with two variables are selected the best who in the next layer are again combined.

Third condition: *There is a selection criterion for validation and measure of the usefulness of an organisation compared with its intended task.*
Since the main purpose of SODM is prediction, it has implemented a criterion of selection that produce powerful predictive models. One most efficient solution here provides the Prediction Error Sum of Squares (PESS) criterion:

$$\text{PESS}= \frac{1}{N}\sum_{t=1}^{N}(y_t - f(x_t,\hat{a}_t))^2 .$$

It is an external criterion but does not require users to subdivide data explicitly since it employs cross-validation techniques internally. Therefore, it is appropriated for under-determined modelling tasks or for modelling short data samples.
Since the selection characteristic of PESS is very smooth, a discriminating criterion, the approximation error variance criterion, is used to make selecting several best model candidates in each layer more certain. After preselecting F^*_p best model candidates from the PESS criterion value, F_p best candidates are selected finally from the preselected set according to the approximation error variance criterion ($F^*_p > F_p$).
In the following is shown, that these principles can be used in the initial data mining process (section 4), but also in several steps of KDD such as estimation of missing values (section 5.1), dimensionality (data) reduction (section 5.2), validation (section 6), and synthesis of alternate models (combining) (section 7).

4. Self-organising modelling

Self-organising modelling consists of several steps of self-organisation, such as

- self-organisation of transfer functions (neurons);
- self-organisation of networks
- for multi-output systems: self-organisation of systems of networks (autonomous systems of equations);

In such a way it is possible to generate from short and noisy data samples

- linear/nonlinear time series models,
- static/dynamic linear/nonlinear multi-input/single-output models,
- systems of linear/nonlinear difference equations (multi-input/multi-output models),
- systems of static/dynamic mutli-input/multi-output fuzzy rules

described analytically in all four cases. More than this, for high noise level, nonparametric models (pattern/cluster) are obtained by Analog Complexing. Using Analog Complexing not only prediction but also cluster analysis (AC clustering) and classification (AC classification) [Müller/Lemke, 00] is possible, which is used in the following (section 5) to automate the data preparation step.

4.1 Self-organisation of transfer functions (neurons)

A GMDH algorithm realises for each created neuron an optimisation of the structure of its transfer function (Active Neuron). Each transfer function f_k is adaptively created by another self-organising process and they may differ one from another by their number of variables used and by their functional structure and complexity. SODM has implemented a complete second order polynomial as default analytical elementary model structure:

$$f(v_i, v_j) = a_0 + a_1 v_i + a_2 v_j + a_3 v_i v_j + a_4 v_i^2 + a_5 v_j^2.$$

This abstract elementary model defines the class of possible models for this level of self-organisation. The arguments v_i, v_j represent all kinds of input data like nonlagged input variables $x_{i,t}$, lagged input variables $x_{i,t-n}$, derivative input variables or even functions or models, e.g., $\sqrt{x_i}$, $1/x_i$, $\sin(x_i)$ or $\log(x_i)$. The true model of every created neuron is instantiated adaptively by self-organisation. As a result, the synthesised network is a composition of different, a priori unknown neurons, and their corresponding transfer function have been selected from all possible linear or nonlinear polynomials: $f(v_i , v_j)$

4.2 Self-organisation of networks

The second level of self-organisation employs a multilayered-iterational GMDH algorithm. There are two enhancements to the basic algorithm, however.

The first difference is that the neurons must not have two input variables due to their self-selecting capability. The second difference of SODM algorithm is applying a so-called layer-break-through structure: all original input variables v_i and all selected F_p best models of all p preceding layers are used as inputs to create the model candidates of layer p+1. The enhanced version breaks open this fixed layer dependence structure, and it allows considering any selected model candidate (or original input variable) as input information at any layer of model creation.

This greater flexibility of model synthesis, however, also amplifies the danger that models are becoming increasingly colinear with growing number of layers. To avoid evolution of colinear input variables generated during modelling, a statistical test is processed before any new neuron will be created excluding colinear input pairs from modelling in this way. Such an algorithm for self-organisation of multi-layered networks of active neurons performs the creation of an optimal complex network structure (optimal number of neurons and number of layers) and selection of a number of best model candidates out of populations of competitive models.

4.3 Self-organisation of systems of networks (autonomous systems of equations)

Complex systems usually have several output variables. The goal of modelling systems of equations using GMDH is to self-organise a model for each output variable and to identify the interdependence structure between the system variables including separating variables into endogenous and exogenous variables according to their corresponding model quality. After modelling a system of m equations, SODM selects a best autonomous system consisting of m* equations (m* < m) necessary to describe the system completely. Here, the number m* of equations the best system consists of and its composition of variables is completely detected by the algorithm using a system criterion. All variables of the identified best system can be considered as endogenous variables of the system. All remaining variables which may be part of the autonomous system are either exogenous or are identified as exogenous due to an insufficent data basis.

5. Self organisation in data preparation

Data preparation builds a ready to model dataset. But preparing data for modeling has been an extremely time-consuming process (nearly 60 % [Pyle, 99]). A first objective in preparing the data set is to prepare the data in such a way that the information content is best revealed for the tool to see. A second objective is to obviate the problems where possible. This techniques can reduce the error rate in a module, reduce modelling time and give enormous insight into the data and therefore is a source of most important benefits.

5.1 Missing values

In most practical applications we have to do with some of the values in the data set not observed. Creating and inserting some replacement value for the missing value the objective it is to guarantee that this values at least are neutral that is neither adds nor subtracts information from the data set. It must introduce no bias. Poorly chosen values adds information to the data set, that is not really present the missing value and in such a way distorting the data. It is necessary to preserve the between-variables relationship, which will be explored in the next modelling step.
Discovering and explication the nature of that relationship is one of the main tasks of the data mining part, called modelling, that comes after preparation. SODM algorithm can be used also in the preparation step to pre-estimate this relationship, which is not actually linear. It has to be understand, that it is not the accuracy of this predictions of missing values that is of most importance when replacing them. The key concern is that the predictions produce a workable estimate that least distorts the values that are actually present. "The purpose of replacing missing values is not to use the values themselves, but to make available to the modelling tools the information contained in the other variables values that are present. If the missing values are not replaced, the whole instance value may be ignored"[Pyle, 99].
With the variables values that are jointly present in the initial sample data set good estimates of missing values of any variable can be made using SODM algorithms. This approach has two steps:

1. deletion of missing values (list deletion)
   All observations with one or more missing values are deleted. On the base of such reduced data set by means of SODM algorithm are generated for every variable with missing values a linear or nonlinear model, where as variables are used all variables of the whole data set without the considered variable including the output variables. Specially for small data sets is convinient the advantages of SODM algorithm generate models on small data sets.
2. Using the generated models for every variable by means of what-if-prediction the missing values can be estimated. After this the whole data set can be used to solve the data mining task.

If there is more than one missing value in one and the same record it may be difficult to estimate the unknown values by regression models, because of conflicts between variables with missing values (interdependence between variables to predict). In this case AC classification is useful. For the given record with missing values by AC classification are selected most similar records which does not contain missing values. After this, the missing values are estimated as mean value of measured values of similar records.

5.2 Data (dimensionality) reduction

A crucial problem is determining how much data is needed for modeling. Reducing the dimensionality, it is necessary to enhance the relationships really in the data. Therefore, the mostly not proofable demand is the data sets needs to be representative. Secondly, a concentration of instances has to enhance the whole information about between variable relationships but also the variability of individual variables. Practical data mining application has to handle with mountains of data, i.e. tables with high dimension. Besides the known theoretical dimensionality problem [Cherkassky, 98] there is a dimension limit of all known tools connected with computer time and working storage. Therefore a step of high priority is the objective choice of essential variables - state space reduction -but also the exclusion of redundant observations - sample space reduction.

5.2.1 State space reduction

a. Modular approach

The modular approach relies on decomposing a given task among a number of networks, each solving a seperate sub-task. Given are N variables x_i, from which M variables y_j has to be predicted. The following approach can be used for dynamic systems:

1. Randomly the variables are grouped in P groups with nearly the same number of variables. Every group gives a subdata set.
2. For every subdata set is generated a system of equations by means of SODM and estimated the system prediction for all variables. If the variables y_j is included in the group, a prediction is obtained, in the other case a model of variables y_j depending of all variables included in the group is generated and by means of what- if - prediction the unknown prediction of y_j evaluated.
3. The obtained P predictions for every variable y_j are combined by means of SODM algorithms.

b. Self-organising variable selection

The basic idea is to use the GMDH principle, it is the aim to reduce high dimensional problems by solving many small problems. The variable set S_0 with a high number of variables is divided into m subsets with equal number of variables. In the first generation for every combination in pairs of subsets linear or nonlinear models are generated. Every generated model contains variables, the set of all variables, contained in $\binom{m}{2}$ models of the first generation give the variable set S_1 of first generation. In the second generation the new variables set S_1 will be divided in equal subsets and for every combination in pairs of subsets again linear or nonlinear models are generated. The whole set of variables, contained in models of second generation gives the variables set S_2 and so on. This has to continue up to a given number of variables is reached .

Obviously, for very high number of variables, such an approach does need many computational effort, the number of models which has to be generated is too much. In this case only a partition in equal groups of variables is useful, but there is the danger, that not all influences in pairs on the output are considered.

5.4.2 Sample space reduction

We have to distinguish stationary processes or homogeneous samples of observations and nonstationary processes or inhomogeneous samples of observations. In both cases we can use self-organising clustering to generate homogeneous or stationary parts of the whole process/sample. But combining of results is possible only in first case. The second case gives models for different situation. Using this model ensemble it needs therefore a classification of the given situation and then application of the corresponding model.

a. Stationary processes

If there is a big number of observations, i.e. a high sample size one useful approach the sample has to be divided by clustering in several clusters of sample observations. After clustering every cluster can represented by one observation. This representative can be selected as

- one specific observation,
- mean value of all obervations in the cluster,
- observation, which has the lowest distance to all other a.s.o.

Principally, it has to be considered the fact that redundancy of samples contains some information. To delete the redundant instances means to lost this information. Therefore it is better to use weights, which give instances with repeated realisations a higher weight than such, which only one times will be.

b. Nonstationary processes:

By means of AC clustering the observations are divided into a small number of clusters with similar records. Every cluster contains observations of a nearly stationary process. After this for every cluster can generated by means of SODM algorithms a special model.

6. Validation

A very important problem in knowledge discovery from data is analysis and validation of the obtained models. This evaluation process is an important condition for application of models obtained by data mining. From data analysis, only, it is impossible, however, to decide whether the estimated model reflects the causal relationship between input and output, adequately, or if it is just a stochastic model of noncausal correlations. In this paper validation means to proof if the derived pattern do exists, actually, and is important for practical applications or if it is only a stochastic one.

By means of Monte Carlo simulation a noise sensitivity characteristic was generated that provides the required external information that helps to decide if the generated input/output model is valid or not. The idea here was building models on a subsequently increasing number of potential inputs M and random samples N several times to get a characteristics for a certain algorithm on how strong the algorithm can filter out noise based on a given data set dimension (N, M). In result, a boundary area Q_u =f(N, M) was obtained that *any* model must exceed to be considered valid to a certain degree of significance in that it reflects relevant relations in the data.

Based on the simulation data of 750 samples and 2 respectively 4 inputs when also using the inverse of N and M) the following model - named in the following test function - was created by "KnowledgeMiner" (fig.1):

$$\hat{Q}_u = \hat{a}\frac{1}{N} + \hat{b}\frac{M}{N}, 0 \le \hat{Q}_u \le 1,$$

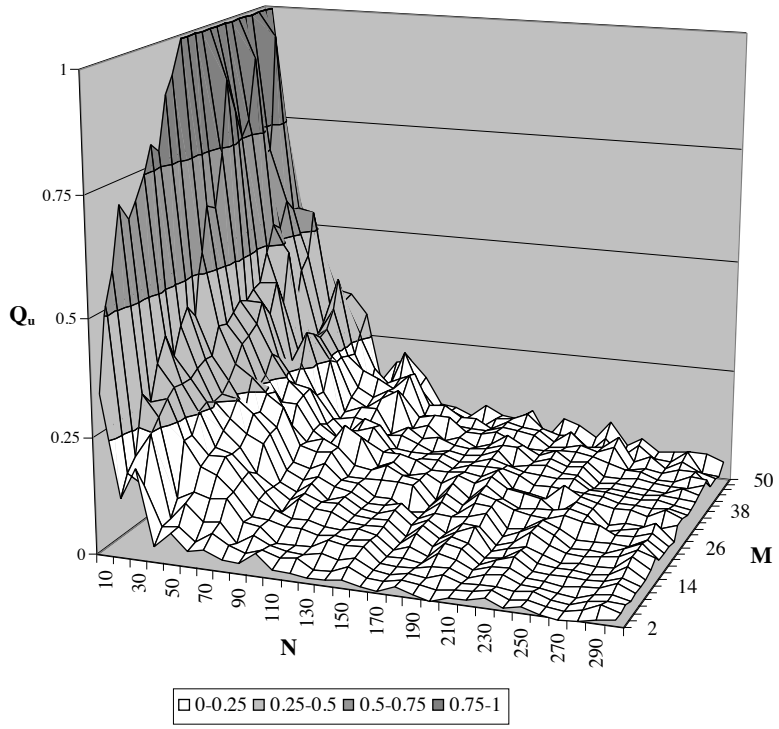

Figure 1: Noise filtering characteristics Q_u

and concluding from that model

$$N \ge f(M \mid Q_u \le \varepsilon) = cM + d, c = \frac{\hat{b}}{\varepsilon}, d = \frac{\hat{a}}{\varepsilon}, \text{ and } M \le f(N \mid Q_u \le \varepsilon) = eN + g, e = \frac{\varepsilon}{\hat{b}}, g = -\frac{\hat{a}}{\hat{b}} \text{ (fig. 2).}$$

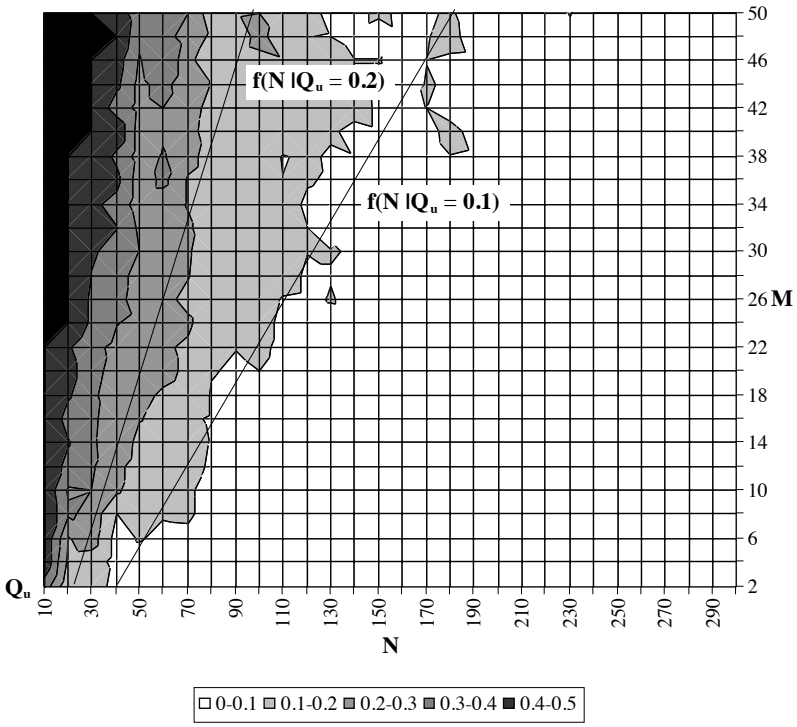

Figure 2: Surface plot of Q_u along with M=f(N) graphs for Q_u=0.1 and Q_u=0.2

To evaluate if the extracted relation Q_u=f(N, M) do extrapolate well, we run the simulation on N, M values that were not included in the data set used for Q_u model estimation. This simulation shows for M=100 and N=10 (50) 810 that theoretical model $Q_u = \overline{Q} + 2s_Q$ and estimated model $\hat{Q}_u = f(N,M)$ are fitting very close, which confirms applicability of the estimated model on the extrapolated parameters.

Concluding from these simulations it seems reasonable that the obtained test function $Q_u(N, M)$ provides a tool that helps to estimate on the fly the validity of a model generated using GMDH. Given a data set of dimension (N, M), a model's quality Q^M can be calculated and compared to a corresponding threshold Q_u. This threshold expresses a "model quality" level that can be obtained when simply using random numbers as a data basis. For a model of a quality $Q^M \le Q_u$, it cannot be verified - due to missing error cases - whether the model reflects some relevant relations or if it just models noise. Therefore, such a model has to be considered invalid. For the other test case, $Q^M > Q_u$, it can be concluded that the probability of the test indicating a model valid for actually non-relevant relations in the data decreases fast asymptotically as the difference $Q^M - Q_u$ rises. This is a most important fact, because, having the error rate available this time , this implies that as ΔQ rises, the probability of testing an actually valid model valid quickly increases to almost 1.

7. Improvement of model results

If the model is not valid

- in the data base are not included most important input variables. Therefore the investigated variable cannot be explained by a input-output model sufficiently. The variable should by considered as exogenous and should be described by a time series model or by Analog Complexing.
- the data base is not well-behaved, i.e. there are more variables than observations. Besides methods of dimensionality reduction quality of model results can be improved by combining.

In many fields, such as economy, there are only a little number of observations which is the reason for uncertain results. The results obtained by models with small sample are in most cases unsufficient. All methods of automated model selection lead to a single best model. On this base are made conclusion and decision as if the model were the true model. However this ignores the major component of uncertainty, namely uncertainty about the model itself. To improve model results artificial generation of more training cases by means of jittering, randomizing a.o. is a powerful way.
Many researches have shown, that simply combining the output of many predictors can generate most accurate prediction that of any of the individual predictor. Theoretical and empirical work [Sharkey, 99] has shown, that a good ensemble is one where the individual networks are both accurate and make their errors on different parts of the input space. Combining the output of networks is useful only if there is disagreement on some inputs, topology, parameter a.o. Combining several identical networks produces no gain.
The task of combining is: given an ensemble of predictors, sought is a prediction by means of voting or averaging (simple, weighted, Bayesian). Combining the corresponding outputs of a number of trained networks is similar to creating a large network in which the trained networks are subnetworks operating in parallel and the combination weights are the connection -weights of the output layer. It is possible to generate a combination of models (synthesis) by SODM algorithms itself. The big advantage of this approach is that automatically by self-organisation is selected the best (voting) or combined some of the best models linearly or nonlinearly.
One problem in creating network ensembles is the following: because the corresponding outputs on the individual networks approximate the same physical quantities they may be highly positive correlated or collinear. Thus the estimation of the harmful weights for combining such networks may be subjected to the harmful effects of collinearity. Collinearity or linear dependency among the corresponding outputs of the component networks may have computational and statistical ill-effects on the estimation of the combination weights, and than can undermine the generalisation ability. In SODM the problem of collinearity is avoided by means of a statistical test before any new neuron will be created and optimising the information matrix after each new layer.

8. Conclusions

GMDH based algorithms and self-organisation can be used to automate almost the whole knowledge discovery process, i.e. models have been created adaptively and data preparation will be self-organised in special missing values are estimated and dimensionality is reduced.
Automated solutions are more or less based on techniques developed in a discipline named "machine learning" as an important part of artificial intelligence. These are various techniques by which computerised algorithms can learn which patterns actually do exist in data sets. They may not be so intelligent as humans but are error-free, consistent, formidable fast, and tireless compared to humans.
Looking at a model quality or model error criterion does not suffice to state a model valid or not, and thus considering it a good model that generalise well. The “closeness of fit” hype is misleading: Even an ideally fitted model can reflect non-causal, i.e., random relations, exclusively, as well as the “worst” fitted model can be the “best” or “true” model. A model’s closeness-of-fit-criterion needs justification with the “working characteristics” of the algorithm it was created with. In this context, this noise sensitivity characteristics provides the required external information to be able stating a model not being valid or being valid the more as the model’s quality Q^M distinguishes from an externally given quality level Q_u ($Q^M - Q_u >> 0$).
The approach is implemented in a prototype of the software "KnowledgeMiner" and is supported by AppleScript.

Literature

Cherkassky, V. F. Mulier: Learning from Data. J. Wiley&Sons. New York 1998.
Müller, J.-A., F.Lemke: Self-Organising Data Mining. BoD Hamburg 2000.
Pyle, D.: Data Preparation for Data Mining. Morgan Kaufman Publ. San Francisco 1999.
Sharkey, A.J.C.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer: London 1999