

Validation in Self-Organising Data Mining

F. Lemke, J.-A. Müller

<http://www.knowledgeminer.eu>

Causality cannot be inferred from data analysis alone. Therefore, validation of input/output models generated by self-organising data mining means to decide if the pattern derived from data does exist, actually, or if it is only a stochastic one. By means of Monte Carlo Simulation a noise sensitivity characteristic for a given algorithm is identified. This characteristic provides external information required to validate any generated input/output model on the fly if it just models noise or not. If a model is stated not valid, there are some alternatives to improve its quality.

1. Introduction

A very important and still unsolved problem in knowledge discovery from data is analysis and validation of generated models. This evaluation process is an important condition for application of models obtained by data mining. From data analysis, only, it is impossible to decide whether the estimated model can reflect the causal relationship between input and output adequately or whether it is a stochastic model with noncausal correlations. Besides the known problems of data-driven approach there is a need to decide whether the model describes a pattern that actually do exist in the data and is important for practical application or if it is a stochastic one (section 2). This paper describes a way to proof model quality empirically (section 3). If the generated model is stated not valid, there are some alternatives to improve the quality of a model (section 4).

2. Data driven approach

The data-driven approach generates a description of a system's behaviour from observations of that system evaluating how it behaves (output) under different conditions (input). This is similar to statistical modelling and its goal is to infer general laws from specific cases. The mathematical relationship that assigns an input to an output and that imitates the behaviour of a real-world system usually has nothing to do with the real processes running in the system, however. The system is not described in all of its details and functions. It is treated as a black box. Obviously, methods of experimental systems analysis cannot solve an analysis of causes of events for complex objects. There are several important facts, which must be underlined.

First of all, the goal of data-driven approach is to estimate an unknown dependency between input (\underline{x}) and output (y) from a set of past observations. Very important is the fact that models obtained in this way are able to represent a relation between input and output, only, if samples of input and output have its values. Unfortunately, every given data set contains only a part of the whole information about the observed system, i.e., the sample is not complete.

Secondly, other factors that are not observed or controlled may influence the outputs. Therefore, the knowledge of observed input values does not uniquely specify the outputs. This uncertainty in the outputs is caused by the lack of knowledge about the unobserved factors. This results, finally, in a statistical dependency between the observed inputs and outputs [Cherkassky, 98].

Thirdly, there is a difference between statistical dependency and causality. Cherkassky [Cherkassky, 98] underlines that the task of learning/estimation of statistical dependency between (observed) inputs and outputs can occur in the following situations or any combination of them:

- output causally depend on the (observed) inputs;
- inputs causally depend on the output(s);
- input-output dependency is caused by other (unobserved) factors;
- input-output correlation is noncausal.

It follows that causality cannot be inferred from data analysis alone; instead, each of the 4 possibilities or their combination is specified and, therefore, causality must be assumed or demonstrated by arguments outside the data [Cherkassky, 98] and cannot be proven by a technical validation process.

With insufficient a priori information about the system to be modelled, there are several methodological problems we have to focus on before applying data-driven technologies. The incomplete - since finite – data base we always use leads to an indeterminacy of the model and the computational data derived from it. Also, the effectiveness of different conclusions drawn from this data by means of mathematical statistics is limited. This incompleteness of the theoretical knowledge and the insufficiency of the data base causes the problems mentioned in [Müller/Lemke,00].

The data driven approach assumes that a given data set contains information that would be of interest if we could only understand what was in it. Some tool is needed that will help to understand the information that is in the data, turning the information enfolded in it into a form that is understandable. Data analysis has to enable human mind to visualize and quantify the relationships existing within data in order to use its formidable pattern-seeking capabilities. Today, the large volume of data is far beyond the ability of humans to handle it. So, automated solutions have been necessary or only possible. Self-organizing data mining (SODM) shares much of the methodology and structure of exploratory data analysis, including some techniques from it, but also adds some new approaches

3. Validation of input/output models

In the following we want to generate by means of Monte Carlo simulation a noise sensitivity characteristic that provides the required external information which helps to decide if the generated input/output model is valid or not. The idea here is 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 $Q_u=f(N, M)$ is 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.

Using this simulation model, two important facts are known a priori:

- Target and input variables are linear independent one each other and, therefore, no causal relationship exist in the data.
- The best and true model that models that data is $y=a_0$, with a_0 as the mean value \bar{y} .

Implementing this true model as the reference model in the R criterion [Müller/Lemke, 00], we obtain:

$$Q_{N,P} = 1 - \sqrt{\frac{\frac{1}{N-P} \sum_{t=1}^N (y_t - y_t^M)^2}{\frac{1}{N-1} \sum_{t=1}^N (y_t - \bar{y})^2}}$$

The expression in the root corresponds to the approximation error variance criterion δ^2 in SODM. This finally means that the approximation error variance criterion can be used to measure the noise filtering capabilities of the evaluated modelling algorithm.

For $N=300$ and $M=50$, in our simulation, we used KnowledgeMiner's GMDH implementation and AppleScript support in the following way:

- ```

For i=1(1)k, k > 10
 For n=10(10)N do
 For m=2(2)M do
 1. create a data set of n(m+1) random numbers
 2. create a linear (nonlinear) GMDH model on that data set

```

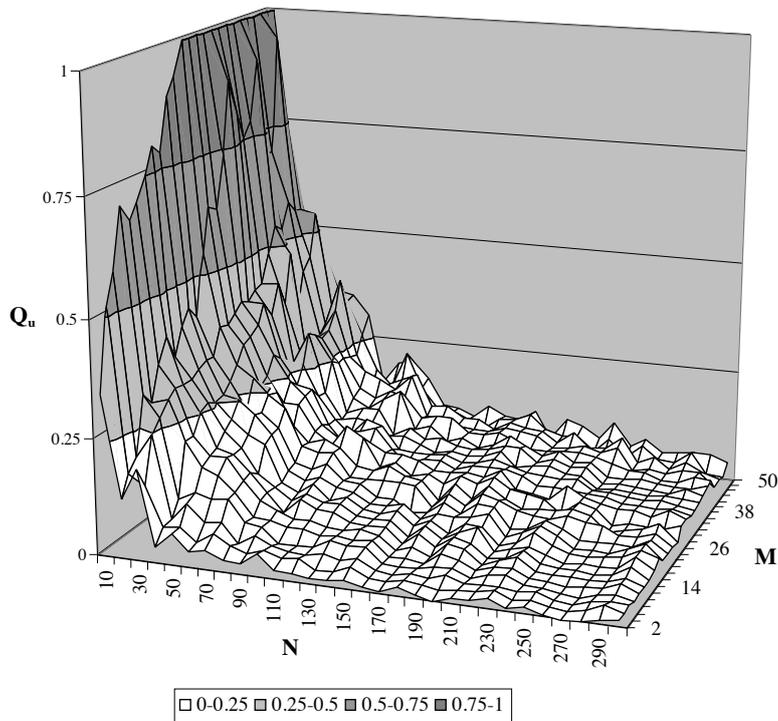
```

3. calculate $Q_{n,m}^i$
end
end
end
calculate $\bar{Q}_{n,m} = \frac{1}{k} \sum_{i=1}^k Q_{n,m}^i$
calculate $s_{Q_{n,m}}(Q_{n,m}^i)$.

```

Running  $k=14$  simulations and given that 750 models are created in each run, a database of 10500  $Q$ -values (models) is generated.

Following the approach suggested in [Müller/Lemke, 00], the empirical noise filtering characteristics of the evaluated GMDH algorithm is given by  $Q_{u,n,m} = \bar{Q}_{n,m} + 2 s_{Q_{n,m}}$  (fig. 1). The value of  $Q_u$  can be seen as a threshold that any model must exceed – or, due to the relation  $\delta^2 = (1 - Q_u)^2$ , the approximation error variance criterion must fall below – in order to be considered valid in reflecting relevant relationships to some extent or significance. Or, in a reverse point of view,  $Q_u$  expresses the probability that a model may also contain non-causal relationships.



**Figure 1:** Noise filtering characteristics  $Q_u$

This plot points out that for some conditions (10 samples and more than 22 inputs, for example) the value of  $Q_u$  reaches the theoretical maximum value of 1 ( $\delta^2=0$ ). This means that in these cases it is not possible at all to state a model valid or not using this tool. It is likely that only prior knowledge about the object that is modelled can help here to make a decision whether or not the model is valid. In other words, knowledge extraction from data should not work for these configurations.

A question of practical importance arising next is: Is it possible to find a mathematical description that models the relation  $Q_u(N, M)$ ? Also, how do  $N$  and  $M$  relate each other given a certain value of  $Q_u$ ? A surface plot of  $Q_u$  suggests a linear relation between  $N$  and  $M$  (fig. 2).

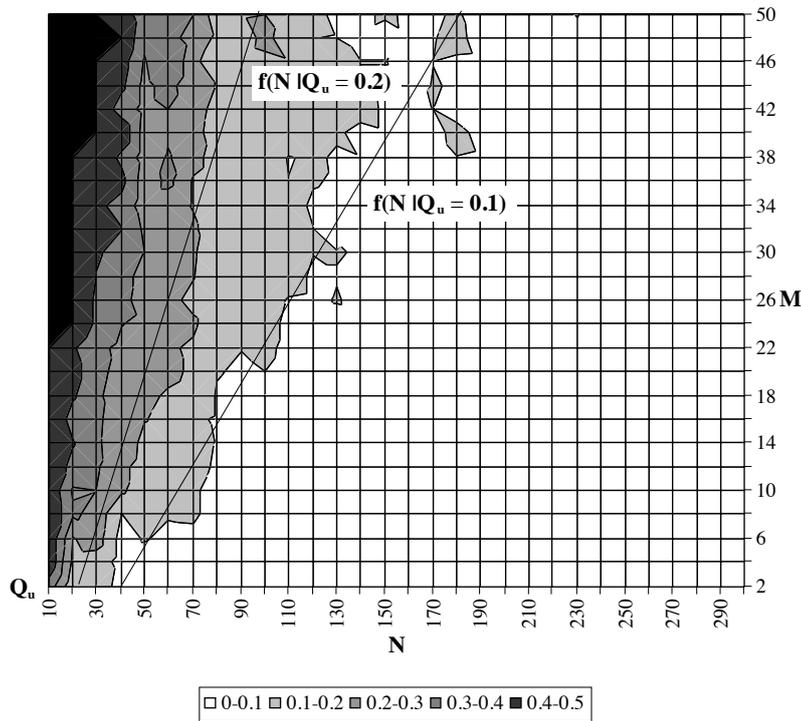
Based on the simulation data of  $Q_u$  (750 samples and 2 respectively 4 inputs when also using the inverse of  $N$  and  $M$ ), the following model - named in the following test function - was created by "KnowledgeMiner"

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

and concluding from that model

$$N \geq f(M | Q_u \leq \varepsilon) = cM + d, c = \frac{\hat{b}}{\varepsilon}, d = \frac{\hat{a}}{\varepsilon}, \text{ and } M \leq f(N | Q_u \leq \varepsilon) = eN + g, e = \frac{\varepsilon}{\hat{b}}, g = -\frac{\hat{a}}{\hat{b}}.$$

The relation  $M=f(N)$  is shown in figure 2 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 the theoretical model  $Q_u = \bar{Q} + 2s_Q$  and the estimated model  $\hat{Q}_u = f(N, M)$  are fitting very close, which confirms applicability of the estimated model on the extrapolated parameters.



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

A contingency table (tab. 1) of the test points out a sensitivity (true positive rate) of 0.972 while the specificity of the test is not quantifiable since no instances was tested that actually reflect relevant relationships. To run a test for causal relations in the data is much more difficult (and generalisable) than running the test for non-causal relations, since the number of instances that represent causal relations is infinite while there is exactly only one true model that reflect the absence of any causal relation:  $y=a_0$ .

|      |                                  | TRUTH                    |                      |
|------|----------------------------------|--------------------------|----------------------|
|      |                                  | Invalid model            | Valid model          |
| TEST | Invalid model ( $Q^M \leq Q_u$ ) | 10206<br>(true positive) | n/a (false negative) |
|      | Valid model ( $Q^M > Q_u$ )      | 294 (false positive)     | n/a (true negative)  |

**Table 1:** Contingency table for testing noncausal relations in data

Looking at the 294 false classified cases ( $\Delta Q = Q^M - Q_u > 0$ ), the error distribution suggests that, as  $\Delta Q$  is growing, the probability to classify an invalid model valid is decreasing to almost zero very quickly. Assuming an exponential error distribution, the probability of  $\Delta Q < 0.052$ , for example, is 0.99, already, which confirms the strong asymptotic behaviour. In the same way, this also means that the probability to classify an actually valid model valid (true negatives) increases to nearly 1, because with

$$\lim_{\Delta Q \rightarrow 1} FP = 0, \text{ FP - false positive cases, it is } \lim_{\Delta Q \rightarrow 1} PVN = \frac{TN}{TN + FP} = 1, TN > 0,$$

where PVN – predictive value negative, and TN – true negative cases.

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" that can be obtained when simply using random numbers as a data basis. For a model of a quality  $Q^M \leq Q_u$ , it cannot be verified - due to missing error cases (false negatives; tab. 1) - whether the model reflects some causal 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 noncausal relations in the data (false positives) 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 (false positives), this implies that as  $\Delta Q$  rises, the probability of testing an actually valid model valid quickly increases to almost 1.

#### 4. Improvement of model results

If the model is not valid then

- a. in the data base are some important input variables missing. Therefore, the investigated variable cannot be sufficiently explained by an input-output model. The variable should be considered exogenous and should be described by a time series model or by Analog Complexing.
- b. the data base is not well-behaved, i.e., there are too much more variables than observations. Besides methods of dimensionality reduction [Müller, 02], quality of model results can be improved by combining.

In many fields, such as economy, there is only a small number of observations available, which is the reason for uncertain results. Results of models obtained on a small amount of samples are in most cases insufficient. All methods of automated model selection lead to a single best model. On this base, however, conclusions and decisions are made as if the model was the true model. This ignores the major component of uncertainty, namely, uncertainty about the model itself. To improve model results, artificial generation of additional training cases by means of jittering, randomization, e.g., might be a powerful way.

Many researches have shown, that just combining the output of many predictors can generate a most accurate prediction. 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 a disagreement on some inputs, topology, or parameters. Combining several identical networks produces no gain.

The task of combining is: Given an ensemble of predictors, a combined prediction is sought 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 the best (voting) or some of the best models is selected and combined 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 colinearity. Colinearity 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, then, can undermine the generalisation capability of the model. In SODM, the problem of colinearity is avoided by means of a statistical test before any new neuron will be created and by optimising the information matrix after each new layer.

#### 4. Conclusions

Philosophically and historically, statistical analysis has been oriented on verifying and validating hypotheses. These inquiries, at least recently, have been scientifically oriented. Some hypothesis is proposed, evidence gathered, and the question is put to the evidence whether the hypothesis can reasonably be accepted or not [Pyle, 99]. In such a way statistical reasoning is concerned with logical justification and like any formal system, not with the importance or impact on the result. This means, in an extreme case, it is really possible to create a result that is statistically significant but meaningless in practical application.

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 be not 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 ( $R^2=1$ ) can reflect noncausal, i.e., random relations, exclusively, as well as the "worst" fitted model ( $R^2=0$ ) can be the "best" or "true" model as this test clearly shows. A model's closeness-of-fit-criterion needs justification with the "working characteristics" of the algorithm it was created with. In this context, a 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  $Q_u$  ( $Q^M - Q_u \gg 0$ ).

#### 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.  
Müller, J.-A. : Self-Organising Data Mining. ICIM 2002. Lvov 2002.  
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