

Does My Model Reflect A Causal Relationship?

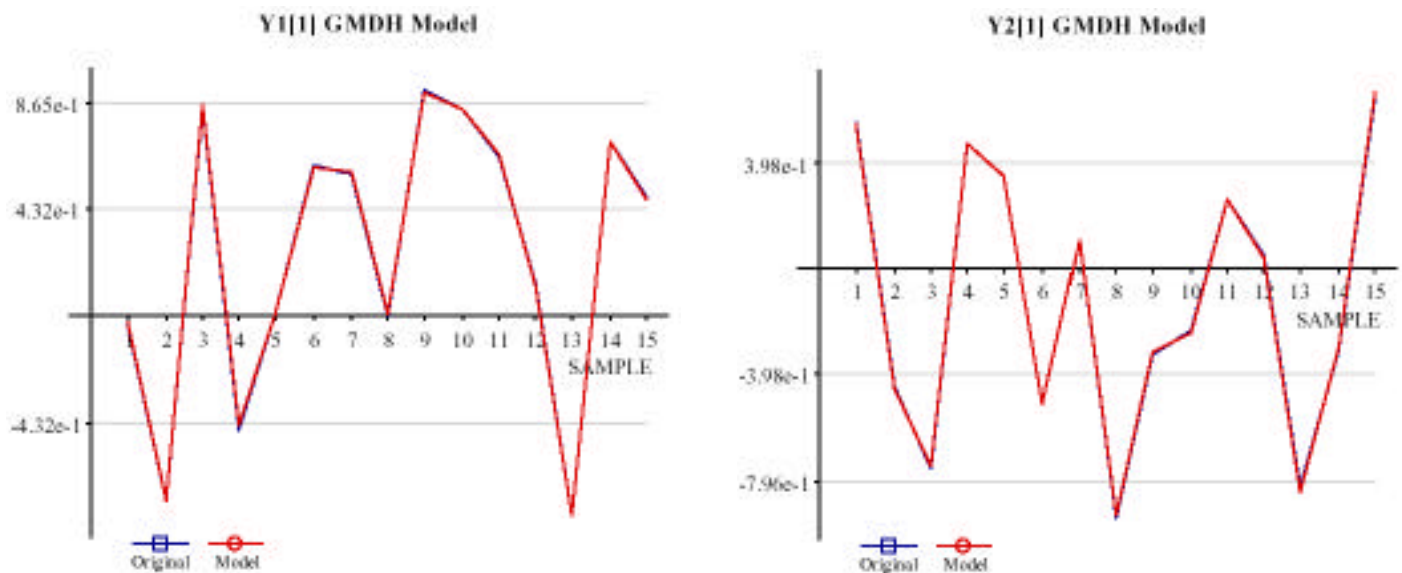
One new feature implemented in the Platinum edition of KnowledgeMiner 5 is evaluation of linear and non-linear GMDH models. This document is about to show how this new model evaluation approach actively supports answering the above question. Also, a new model quality measure that takes into consideration the noise filtering power of the algorithm and model complexity is introduced: Descriptive Power.

The Problem

A key problem in knowledge discovery from data is final evaluation of generated models. This evaluation process is an important condition for application of models obtained by data mining. From data mining, only, it is impossible to decide whether the estimated model can reflect the causal relationship between input and output, adequately, or if it's just a stochastic model with noncausal correlations. Model evaluation needs - in addition to a properly working noise filtering for avoiding overfitting the learning data - some new, external information to justify a model's quality, i.e., both its predictive and descriptive power.

Why

Let's have a look at this example: Based on a data set of 2 outputs and few inputs, KnowledgeMiner creates a GMDH regression model for each output variable Y1 and Y2 (fig.1).



a) Model 1: $Y1=f_1(x)$

b) Model 2: $Y2=f_2(x)$

Fig. 1: Model graph of two models

For model 1, a Coefficient of Determination (R^2) of 0.9998, an Approximation Error Variance (AEV) of 0.0002, and a cross-validated Prediction Error Sum of Squares (PESS) of 0.0005 is reported, while model 2 shows a R^2 of 0.9997, an AEV of 0.0003, and a PESS of 0.0006. Concluding from these or any other common model quality or error criteria and from the graphs of fig. 1 there is no reason to not classify both models as "true" models that reflect a causal relation between output and input. Also, remembering that a most important difference of KnowledgeMiner compared to the vast majority of data mining tools is its inductive, self-organized model synthesis that implements a powerful noise filtering during modeling, already (see also "Self-Organising Data Mining" [book](#), section 3.2), this underlines the above assumption.

However, the person who created the data set for this example states that only one model actually describes a causal relationship while the other model simply reflects some stochastic correlations, because output and inputs are completely independent one another (random numbers). Even with this information given - which is usually not the case for real-world data - the modeler cannot decide from the available information which of the two models is the true model. Only applying the models on some new data (which adds new information) will turn out the true model (fig. 2):

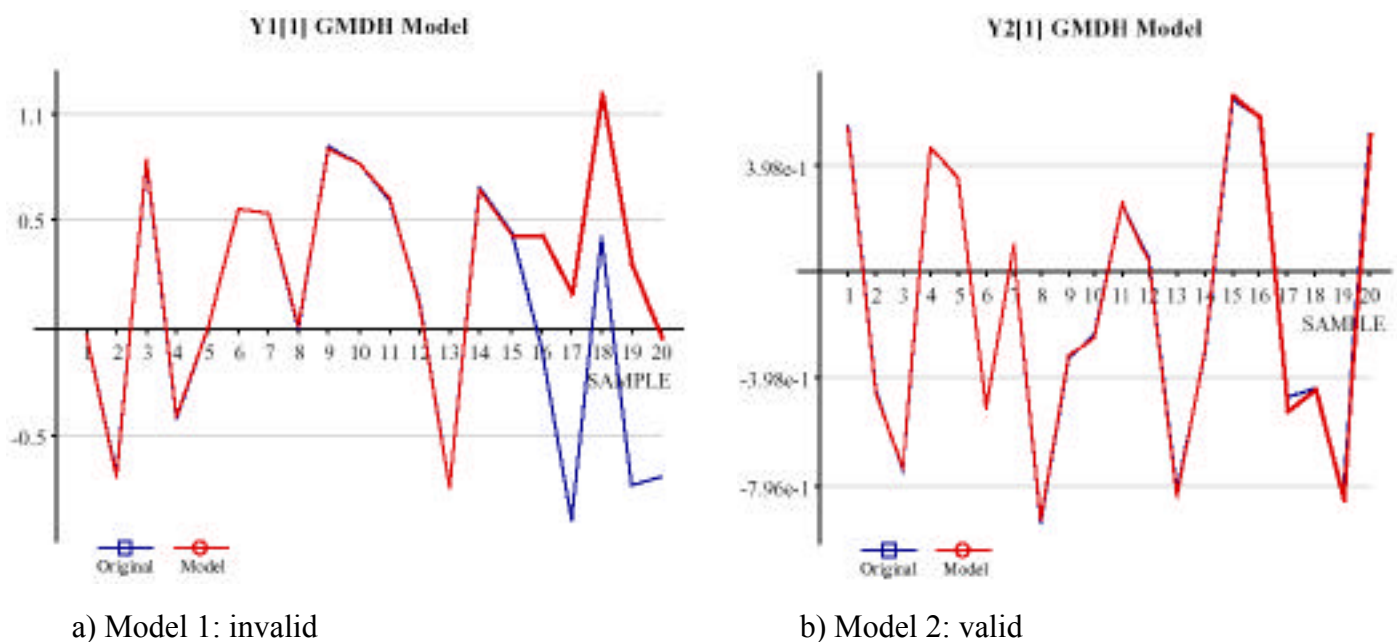


Fig. 2: Prediction of the two models

This example clearly shows that any "closeness-of-fit" measure does not suffice to evaluate a model's predictive and descriptive power, finally. Recent research has shown that model evaluation requires a two stage validation approach (at least):

1. Level

Noise filtering to avoid overfitting the learning data (hypothesis testing) based on external information not used for creating a model candidate (hypothesis) as an integrated part of the "Learning" process. A corresponding tool that has been using in KnowledgeMiner from the beginning within "Learning" is leave-one-out cross-validation, expressed by the PESS criterion.

2. Level

A characteristic that describes the noise filtering behavior of the "Learning" process to justify model quality based on external information not used in the first validation level yet. An approximation of

the noise filtering characteristic is implemented in KnowledgeMiner 5 Platinum for the first time for linear and nonlinear GMDH models. This characteristic was obtained by running a Monte Carlo simulation many times, so it expresses a kind of new, independent "common knowledge" that *any* model can be and must be adjusted with (see also "[Validation in Self-organising Data Mining](#)"). Figure 3 shows a detail of the characteristic of linear GMDH models.

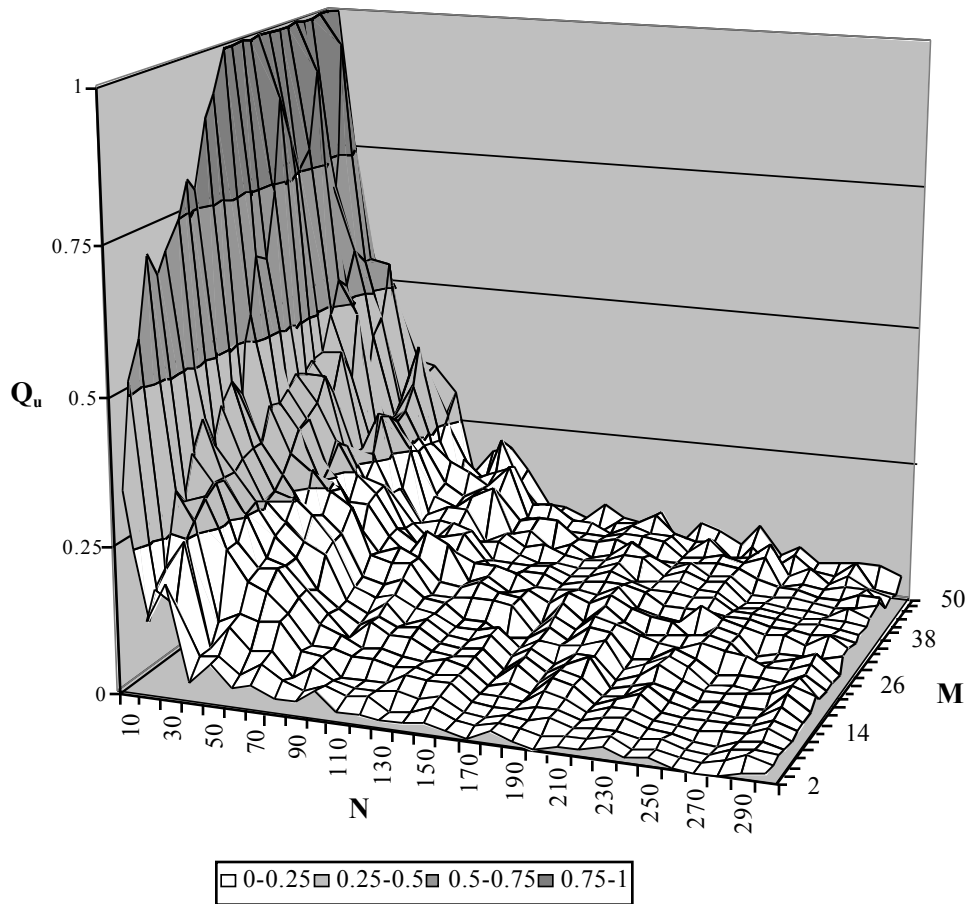


Fig. 3: Noise filtering characteristic

M: number of inputs; N: number of samples; Q_u : virtual quality of a model

$Q_u=1$: noise filtering does not work at all; $Q_u=0$: ideal filtering

The reason for a second level validation is (1) that noise filtering implemented in level 1 is very likely to not being an ideal noise filter and thus not working properly in any case (see example) and (2) to get a new model quality measure that is adjusted by the noise filtering power of the algorithm.

The noise filtering characteristic expresses a virtual model quality Q_u that can be obtained when using a data set of M potential inputs of N random samples. It is virtual model quality, because, by definition, there is not any causal relationship between stochastic variables (true model quality $Q=0$, by definition; for a definition of model quality Q, please see "[Validation in Self-organising Data Mining](#)"), but there are actually models of quality $Q > 0$, which, when using random samples (see example above), just reflect stochastic, nonexistent correlations. In result, given any number of potential inputs M and number of samples N, a threshold quality $Q_u=f(N, M)$ can be calculated by KnowledgeMiner that *any* model's quality Q must exceed to be stated valid in that it

describes some relevant relationship between input and output. Otherwise, a model of quality $Q = Q_u$ is assumed invalid, since its quality Q can also be reached when simply using independent variables, which means that this model does not differ from a model of just stochastic correlations. It's simply garbage.

In addition to deciding if a model appears being valid or not, the noise filtering characteristic is also a tool for quantifying to which extent the data is described by a causal relationship between input and output. This introduces a new, noise filtering and model complexity adjusted model quality measure: *Descriptive Power* (DP), which is defined as:

$$DP = \begin{cases} 0 & Q = Q_u(N,L) \\ \frac{Q - Q_u(N,L)}{1 - Q_u(N,L)} & Q > Q_u(N,L), Q_u(N,L) < 1 \end{cases}$$

with Q as the measured quality of the evaluated model and $Q_u(N, L)$ as the reference quality calculated from the number of samples N the model was created on and from the number of input variables L the model is actually composed of (selected relevant inputs), with $L \leq M$. This means that Descriptive Power is corrected by any virtual quality that may exist and that directly allows for model complexity. For example, two models M_1 and M_2 show the same quality $Q = Q_1 = Q_2$, but M_1 uses more relevant inputs than M_2 to reach that quality Q , so, with $L_1 > L_2$, the Descriptive Power of M_2 is higher than that of M_1 .

The bottom line

KnowledgeMiner 5 Platinum now evaluates a created model and calculates its Descriptive Power on the fly. You don't have to care about it. KnowledgeMiner will serve you with all corresponding information in the model report to make you more effective and successful in your data mining and knowledge extraction efforts.

Back to our example above, KnowledgeMiner 5 Platinum will provide this additional information in the report for the two models (fig. 4):

MODEL EVALUATION:

The model cannot be validated, because noise filtering does not work on this unsufficiently sized information basis. The obtained model accuracy can also be reached when just using random numbers as input data.

Increase the number of samples to about 75 and/or
decrease the number of potential input variables to below 3.

a) Report of Model 1 --> status: uncertain

MODEL EVALUATION:

The model appears to reflect a valid relationship. It describes 98 % of the data.

For the chosen model type, modeling is based on a VERY POORLY sized information basis, which may result in a not properly working noise filtering. To improve the noise filtering capability of the algorithm and thus improving model reliability, it is highly recommended to decrease the variables/sample ratio. If possible increase the

Number of Samples: to above 131 and/or decrease the
Number of Variables: to below 3.

b) Report of Model 2 --> status: valid

Fig. 4: Reported evaluation results of the two models

In result of modeling we now get the information that model 2 really does very well (DP of 98%), while model 1 is uncertain. Trying to follow the recommendations given in the report of model 1 and decreasing the number of potential inputs to 4, in a second modeling run, KnowledgeMiner comes up with this report (fig. 5):

MODEL EVALUATION:

The model seems to reflect a random relation only. Either the data used is actually noncausal or the noise-to-signal ratio of the data is too high to validate a relevant relation. If the data is considered noncausal, the best model that describes the data is just the mean value of the target variable: $y = 0.2469$.

Fig. 5: Evaluation result of model 1 after remodeling --> status: invalid

This result, which is the true result, is the more interesting as the model still fits the data quite well ($R^2=0.8173$ see also model graph in fig. 6; would you have considered this model invalid when viewing that graph?), however, KnowledgeMiner correctly calculates a Descriptive Power of 0% for this model. In only two steps, and using KnowledgeMiner 5 Platinum on some training data only, we have learned that model 2 is a valid model that will generalize well while model 1 simply pretends a high quality and a relation that is not true in reality, and therefore, has to be rejected.

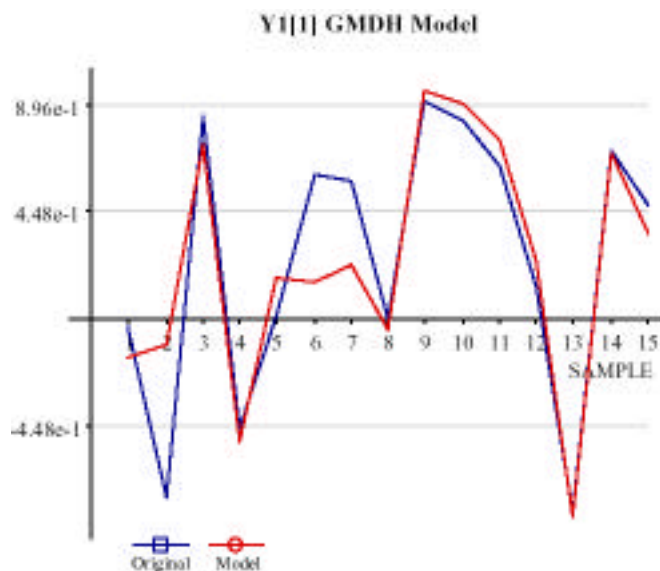


Fig. 6: Model graph of model 1 after remodeling --> looks good, but the closeness-of-fit is misleading: A Descriptive Power of 0% correctly indicates the true noncausal nature of the data

The implemented two stage model validation approach now allows, for the first time, to get an active decision support in model evaluation for minimizing the risk of false interpreting models and using invalid models that just reflect some noncausal correlation.

We appreciate your [feedback](#) about your experience and results from with or without using (other tools) this feature.