Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

VISUALIZATION TECHNIQUES UTILIZING THE SENSITIVITY ANALYSIS OF MODELS

Ivo Kondapaneni
Pavel Kordík
Pavel Slavík

Department of Computer Science and Engineering, FEE
Czech Technical University in Prague
Karlovo náměstí 13, 12135 Praha 2, CZECH REPUBLIC

ABSTRACT

Models of real world systems are being increasingly generated from data that describes the behaviour of systems. Data mining techniques, such as Artificial Neural Networks (ANN), generate models almost independently and deliver accurate models in a very short time. These models (sometimes called *black box* models) have complex internal structures that are difficult to interpret and we have very limited information about the credibility of their output. A model can be trusted just for certain configurations of input variables, but it is hard to determine which output is based on training data and which is random. In this paper, we present visualization techniques for exploration of models. Primary goal is to consider the behavior of the model in the neighborhood of the data vectors. The next goal is to estimate and locate the ranges in input space where the models are credible. We have developed visualization techniques both for regression and classification problems. Finally, we present an algorithm that is able to automatically locate the most interesting visualizations in the vast multidimensional space of input variables.

1 INTRODUCTION

The real world is a very complex system, which is driven by unknown (currently) laws. These laws are what people want to describe, to be able to perform predictions or approximations of the real world's behaviour.

When we want to model the real world, we must firstly describe it's behaviour in terms of input and corresponding output (behavioral analysis). Input and output are always taken as N-dimensional vectors of scalars, and each part of these vectors can be measured in some quantity.

Then we must select and develop an appropriate model type, which will estimate real world behaviour. This developing process consists from learning of model upon given measured data. After this learning process, the model should be able to return an output for the given input, which is an estimation of the real world's response to the same input.

We measure the input and output data, and these pairs serve to us as a training (and/or testing) dataset for the model creation process. ANNs trained on such data should return the right answer on any given input vector from the measured data, but also they give an answer for an arbitrary input vector (we say that they have generalization capability).

One question which we often deal with is: How is the output of the artificial model related to the measured input? Well-trained models should have the correct response to every input from the measured dataset, but we are also interested in how the model generalizes. If the model generalizes in the right manner it can be claimed as credible, therefore revealing some knowledge about the world (eg. knowledge about how the set of input features is interrelated and how input is related to output). And here we need some mechanism for the model's exploration together with measured inputs and outputs. And, in turn, we are dealing with the problem of the dimensionality of input space.

In our work we concentrate on the visualization of such models in order to obtain better knowledge about them and perform some credibility analysis. We also strive to find some interesting areas in model (i.e. areas where the model's output changes rapidly in respect to input).

There are not many works dealing with the visualization of the black box model's output. The main problem is the dimensionality of the domain space of black boxes. Feature space often consists of more than three features, and the main problem is seeing if the model is credible and has a reasonable output. Also, there are a lot of publications about visualizing multidimensional data, but only a minority of it deals with the problem of visualizing the model's behaviour itself.

A very large survey of visualization techniques used up to now can be taken from a lecture by Keim (1997). Here we can find commonly known techniques, such as

parallel coordinate plots, *scatter plots* or *projection methods*, along with techniques for interacting with visualizations. Distortion techniques are also often used, but we recognize much of them as too confusing for the experts, so it is better to represent the dimensions as they are.

We are mainly concerned with geometric techniques, so we were aimed in this direction in our research. One geometric approach for dealing with multidimensional functions in their visualizations is to slice and project multidimensional space into 3D/2D space. Many solutions have emerged in this field; for example Santos and Brodlie (2002) developed *hypercell slicing*. This technique is designed in such a way that users create multidimensional workspaces (i.e. region of interest) and in these workspaces they create 3D slices (i.e. projections) from multidimensional space. Authors have claimed that the solution is practical, but it is commonly known that such techniques need a lot of user attention in order not to get lost in the multidimensional space.

Instead of visualizating multidimensional functions by projections, some authors (Jardins and Rheingans 1999) explored the possibility of exploring artificial models trained over some data by mapping multidimensional space into 2D space. They used *Self-Organizing Maps* and their solution proved to be quite interesting; they enable users to perform tasks like testing model confidentiality and comparison of several models. The only thing we must accept when we use such maps is that we lose the sense of cartesian space, with which we are familiar and it may cause confusion for the experts.

Also we try to use visualization to reveal model sensitivity with respect to inputs, so we want to perform model sensitivity analysis in some form. There are two practical options for performing such a task: direct observation of the model or some kind of computation.

Sensitivity analysis through direct observation can give us very good information, but it is very hard to find where the interesting regions are positioned (especially in more than three dimensions). Such approaches are often combined with some type of automatic exploration mechanism (as we will see in the description of our solution).

Some survey of sensitivity analysis methods based on computation can be found in M. Gevrey (2003). There are mainly methods, that were originally developed for ANNs, but some of them can be used directly for any black-box model analysis. Among the tested methods are the classic *partial derivatives method* (PaD) or *Perturb* method. Authors have compared methods to each other and concluded that very good results were offered by *PaD*, and as the second most informative, the *Profile* method, was introduced.

Also, we find it interesting when authors deal with the visualization of computed sensitivity for some ANN, as in Theron and Paz (2006). Their solution consists of completing sensitivity analysis in areas which are positioned at training points. For each training point, they find the sensitivity of the black-box model in each dimension (i.e. input) of data. These results are then represented as some kind of stacked-bar graph, where each column represents the sensitivity analysis for one training point. They also use parallel coordinates, which serve as some additional tool for analysis. Their solution is not oriented only towards ANN; it is also *black-box* oriented.

Some authors (Tzeng and Ma 2005) have developed a method of visualizing input data along with an ANN inner structure. As a result, we can see an artificial network with its topology, but augmented by the importance of inputs and the interconnections between the layers. Therefore, experts can directly modify their ANN to achieve a simpler network or to better understand it.

In our research, we are also interested in the ensemble of models (Hansen and Salamon 1990) also called model committees (Abdel-Aal 2005). These are frequently used to improve the accuracy of weak learners (decision forests, nearest neighbor techniques, etc.). Improved accuracy is not the only task where the combination ensemble models can help. Ensembles of various models in climatology were also recently used to estimate the credibility of predictions (Stocker 2003). We have several model outputs and the end result is obtained by combining these. What we want to know is where such a combination is credible. This question is solved by finding regions where the output of partial models is somehow similar to the others. As we will see next, such a search can be performed directly by observation or by a genetic search over the model's feature domain.

This paper is organized as follows: In the next section we present how the ensemble of models can be used to estimate the credibility of models. We propose visualization techniques that allow us to study the relationships between variables for regression models and classification boundaries for classifiers. In the last section, we propose a solution that shows how interesting plots can be located in vast multidimensional space. This was achieved by a genetic algorithm with a specially designed fitness function that defines the "interestingness" of particular plots.

2 PROPOSED APPROACH

As was previously stated, when a single model is generated from a data set, it usually reacts well to data similar to that used for training. To compute the accuracy of this model, we can use a so-called testing data set, a collection of data measured together with a training set, but not used in the model's learning phase. Then we statistically evaluate the output error of our model compared to the original measured output (we can use, for example, cross-validation or simple percentage split).

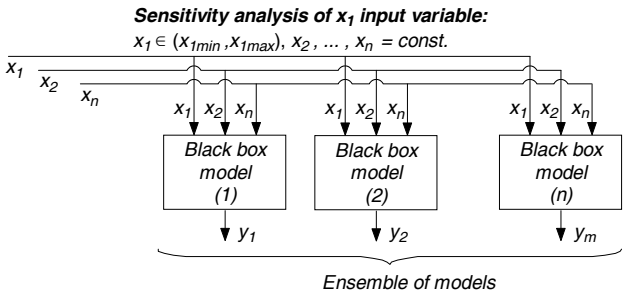

Figure 1: Sensitivity analysis of input variables can be performed according to the picture. We change the value of the corresponding input variable from minimum to maximum whereas other input variables stay constant. The output variable shows sensitivity to the input variable in the configuration of other variables.

The problem with this approach is that the information about the model's accuracy on the testing data is just an indicator of the generalization abilities of the model. It does not say anything about the credibility of the model for particular regions of the input space, that were not covered by the testing dataset.

Our approach exploits the visualization of model ensembles (Zhou, Wu, and Tang 2002) to estimate credibility for any combination of input variables. In our visualizations, we adopt the principle of sensitivity analysis (see Figure 1). Selected input variables are varied from minimum to maximum, whereas other inputs stay constant. Sensitivity (output) of the ensemble to the selected variable is recorded and, together with input data records, provide source data for the visualization techniques proposed below. Then we can directly see in the visualization, how the output of particular models correspond to each other and make conclusions about credibility regions for some point in space.

Our visualization techniques can be used to estimate the credibility of any black-box models, providing that the models in the ensemble are diverse enough.

The ensemble of models used in sensitivity analysis can be built, for example, by means of the Group of Adaptive Models Evolution method (GAME). This method generate hybrid inductive models consisting of a network of interconnected units similar to artificial neurons. To learn more about this method, please refer to Kordík (2006).

In the next two sections, we show concrete visualization techniques and how they can be used. Both utilize the ensemble of models to estimate credibility of the (a) output prediction and (b) classification, depending on the task.

2.1 Visualization of Predictors

The first method is aimed at predictors built to estimate a single continuous target variable. Sometimes these models can additionally be referred to as regression models. The visualization plots in Figure 2 consist of sensitivity responses from nine models (member models of the ensemble) drawn as curves. In the right plot, there are also input data vectors displayed as crosses. The size of individual crosses is computed from their Euclidian distance from the plot (from constant inputs). Curves should pass through the centers of the biggest crosses (in case of data without noise). The plot is, in fact, a one-dimensional slice of the multidimensional input space.

The ensemble of models tries to predict the output variable y in the artificially defined problem $y = \frac{1}{2}\left(\sinh(x_1 - x_2) + x_1^2(x_2 - 0.5)^2\right)$, where training data vectors are distributed uniformly only in the area x_1,$x_2 \in \langle 0,1 \rangle$. The plots in Figure 2 show the sensitivity of the ensemble models to x_1 (left plot) and x_2 (right plot) input variables.

Note that the behaviour of the ensemble models is similar in the area covered by the training data, and increasingly random outside of this area. This information is utilized later to estimate the credibility of the prediction.

The application of the proposed technique to a real world data set (Boston housing data UCI) is in Figure 3. The left plot identifies the negative influence of criminality (CRIM input variable) to the value of houses (MEDV). The visual information can be interpreted with the following question: What will happen to the value of houses if the criminality in the region increases or decreases? The relationship is valid just for a specific house defined by constant values of all other input variables, except the CRIM variable. However, our experiments with several real world data sets showed that the relationship is often very similar in the whole input space (i.e. for all possible houses).

Again we can see that the curves of particular models disperse in regions not covered by the training data, which indicates low credibility of the models in those regions.

The right plot in Figure 3 shows the variable RM (average number of rooms per dwelling) that was considered irrelevant by all ensemble models of the house value. An increase or decrease in its value would not change the value of house (according to models).

It is apparent from the given examples, that with this visualization we can easily find credible regions of the input space together with information about the relationships between the output and selected input variables (in some particular point of input space). This technique can be also easily extended to three dimensions (two input variables are selected and changed from minimum to maximum to generate the 3D plot). You can see an example of 3D plots in Figures 6 and 8.

2.2 Visualization of Classifiers

The second method can be used for ensembles of models (classifiers) built to distinguish a class of data vectors.

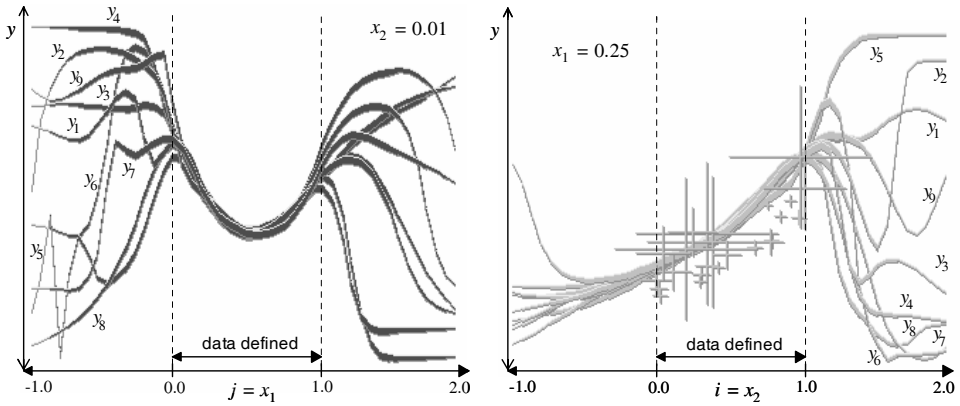

Figure 2: Individual models behave randomly for input configuration where the models were not trained properly. The compromised response of the ensemble signals well-trained models and credible output for the input configuration

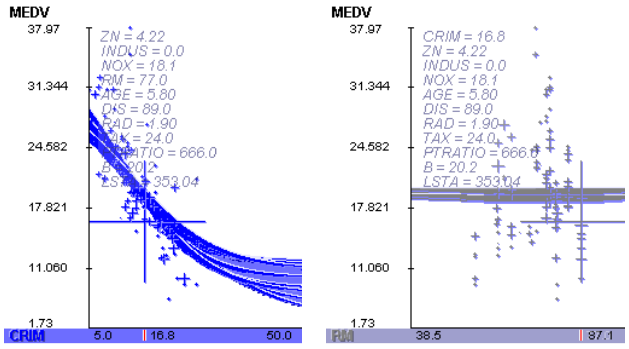

Figure 3: The value of houses (MEDV) predicted from several input variables (Boston housing data set). The CRIM input variable (criminality in the region) has clearly negative influence on the value of houses. The RM input variable feature is considered irrelevant by the ensemble of models.

Again an ensemble of models (classifiers) is built for a single output variable (membership to a particular class). Each model should give "1" on the output, when it classifies patterns of his member class, and "0" when the input vector belongs to another class. For regions far from training vectors, the output of the model is random. For classifiers with logistic transfer functions, the random output is usually close to either "1" or "0".

Let us explain the idea with a simple example. Consider the data about apples and pears. If we evolve the ensemble of classifiers for the apple class, their outputs are "1" for objects similar to apples, "0" for those similar to pears. For an object that is different from both apples and pears, each model from the group can give a different output. Some can classify it as an apple ("1"); some can respond that it is not an apple ("0"). When the outputs of all the models are multiplied, the result is "1" just for objects classified as an apple by the whole ensemble. This simple idea is extended below to filter out artefacts and unimportant information and indicate just credible regions of class membership.

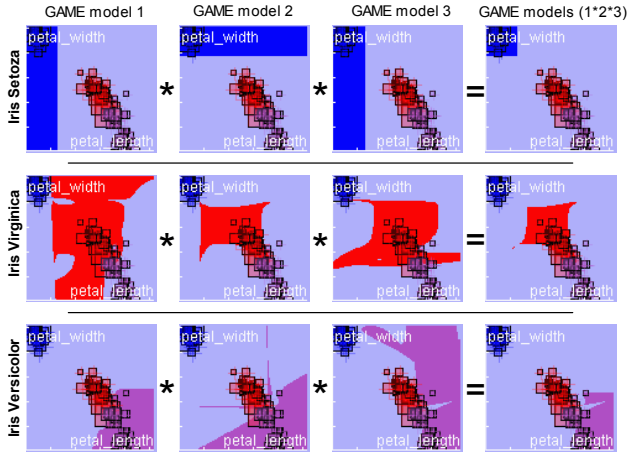

Figure 4: When we multiply responses of several ensemble models of the same class, we get the membership area just for those configurations of inputs, where the output of all models is "1".

This multiplication of classifiers is demonstrated in Figure 4. In this case, two selected input variables vary from minimum to maximum (others stay constant) and the output of the classifier (class) is encoded as color. Dataset is also added into the plot, each data vector is represented as a square of variable size and color. The size of the square is again derived from the euclidean distance of the data vector from the position in the plot. The color corresponds to the class assigned to the data vector.

The Iris data set (UCI) is often used to test the performance of classifiers. Iris plants are to be classified into three classes (Setoza, Virginica and Versicolor) given measurements of their sepal width, length and petal width and length. We developed three ensembles of classifiers - one ensemble for each class. Figure 4 shows three models from each ensemble. A dark background signifies "1" on the output of the model, a light background means an output of "0". When these three models are multiplied for each

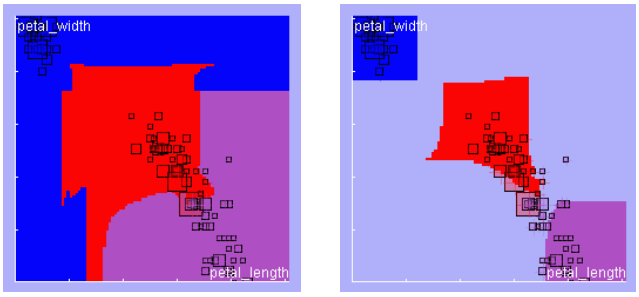

Figure 5: Three groups of twelve GAME models for classes Iris Setoza, Virginica and Versicolor. When all models are displayed in one scatterplot (left), the regions of class membership overlap. The right scatterplot shows the proposed improvement where the outputs of the models within one group are first multiplied and then the result for each class is displayed.

class, the results can be observed in the scatterplots of the fourth column. Each resulting scatterplot has much better defined classification boundaries and does not contain random information.

On Figure 5, you can see how the proposed method improved the classification. The outputs of twelve ensemble classifiers for each class are displayed in one scatterplot (left). Especially plants more distant from those present in the training data are classified as members of several classes. When the twelve models for each class were multiplied first and then the results for the three classes were displayed into one scatterplot (right), the boundaries of membership areas became clearly visible.

This visualization technique can be also extended to three dimensions. The decision boundaries of classes can be for example approximated by NURBS curves (Roger 2001).

3 AUTOMATED RETRIEVAL OF INTERESTING PLOTS

Extending the proposed visualization techniques to 3D space is particulary useful when the modelled data has more than one very significant input.

In the case of predictive models we select two inputs instead of one. Figure 6 shows the sensitivity of a single model trained on the Housing data set used in the section above.

3.1 Genetic Algorithm to Locate "Interesting" Plots

It is often the case that an output variable is sensitive to just a few of the input variables and in a limited range. A manual search for these "interesting" regions in multidimensional input space is very time consuming. Therefore we use a

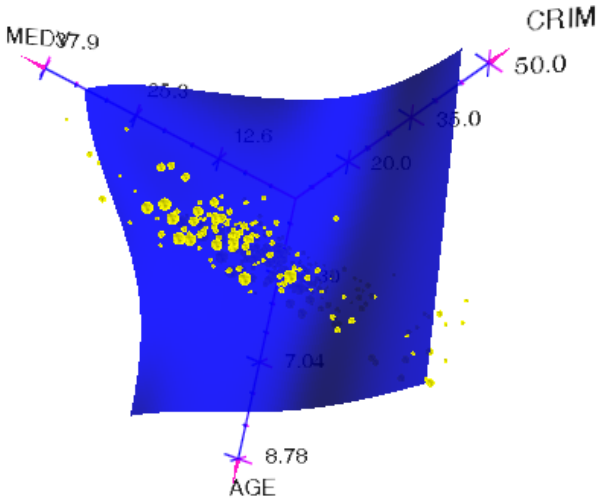

Figure 6: 3D visualization of predictive model - sensitivity of house value (MEDV) to criminality in the region (CRIM) and proportion of owner-occupied units built prior to 1940 (AGE).

genetic algorithm (Goldberg 1989) to locate these regions automatically.

When you look at Figure 7, you can see how we define the "interesting" plot. Each curve represents the response of one ensemble model y_i to the change of x_i variable. We derive the interesingness of a plot from three criteria: maximal difference in response among models (y_{size}), how diverse they are in response (p) and how big their credible area (x_{size}) is. To make these criteria meaningful, we compute them only in areas of the plot, which can be claimed as credible, which means range $\langle x_{start}, x_{start} + x_{size} \rangle$.

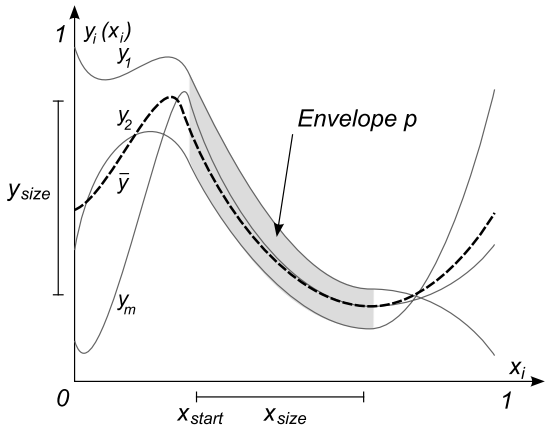

Figure 7: This figure demonstrates "interesting" behavior of models. The relationship is very steep, meaning that the output reacts significantly to the change of input variable. Models in the ensemble have at the same time very similar behavior and the region is big enough to be able to get an insight into the variables relationship.

The first criterion y_{size} can be interpreted as: the bigger change in response we observe in the plot, the more interesting the plot is for us. It is computed as follows:

$$y_{size} = \arg\max\left(\bar{y}(t)\right) - \arg\min\left(\bar{y}(t)\right), t \in (x_{start}, x_{size}), \quad (1)$$

where the term $\bar{y}(t)$ is a Simple Ensemble of models. Zhou, Wu, and Tang (2002) defined as $\bar{y}(t) = \frac{1}{m}\sum_{i=1}^{m} y_i(t)$ for $t \in \langle 0,1 \rangle$.

The second criterion reflects how diverse models are in the plot. The value of this criterion will be small for plots where models are not very diverse, and their response is tightly corelated. The criterion can be computed as:

$$p = \sum_{j=x_{start}}^{x_{start}+x_{size}} \left(\arg\max_{0<i\leq m}\left(y_i(j)\right) - \arg\min_{0<i\leq m}\left(y_i(j)\right) \right), \quad (2)$$

where y_i is the response of i-th model.

The last criterion x_{size} helps us to privilege bigger areas of interesting behavior.

To find the best form of the fitness function, we had been experimenting with several equations (Kordík, Saidl, and Šnorek 2006). The best properties showed the fitness function with three components multiplied:

$$fitness = y_{size} * \frac{1}{p} * x_{size}, \quad (3)$$

The genetic algorithm works as follows. It generates several plots with a random configuration of constant inputs. Each plot is one individual in the initial population. In the next step, the fitness of each plot is computed according to Equation 3. The roulette wheel selects plots that are crossed and mutated to make up the next generation of plots. Generation after generation the proportional fitness of plots increases and after several epochs, the best individual is selected as the most interesting plot.

We validated the functionality of the genetic algorithm on synthetic and also real world data (Kordík, Saidl, and Šnorek 2006). The idea can also be easily extended for 3D plots.

Figure 8 shows a random 3D plot from the initial population of the genetic algorithm and also the most "interesting" 3D plot with highest fitness - the result of the genetic algorithm that located better values of constant input variables that are not used for sensitivity analysis (just two selected input variables are changed in their values to generate the 3D plot). The output of each model is represented by a different color. It helps to visually distinguish areas where models differ and where their output is similar.

By using this method we can automate the process of data analysis. As an practical example we can mention Kordík (2006), where an ensemble of models, together with the best visualizations, are evolved automatically and presented to the domain expert.

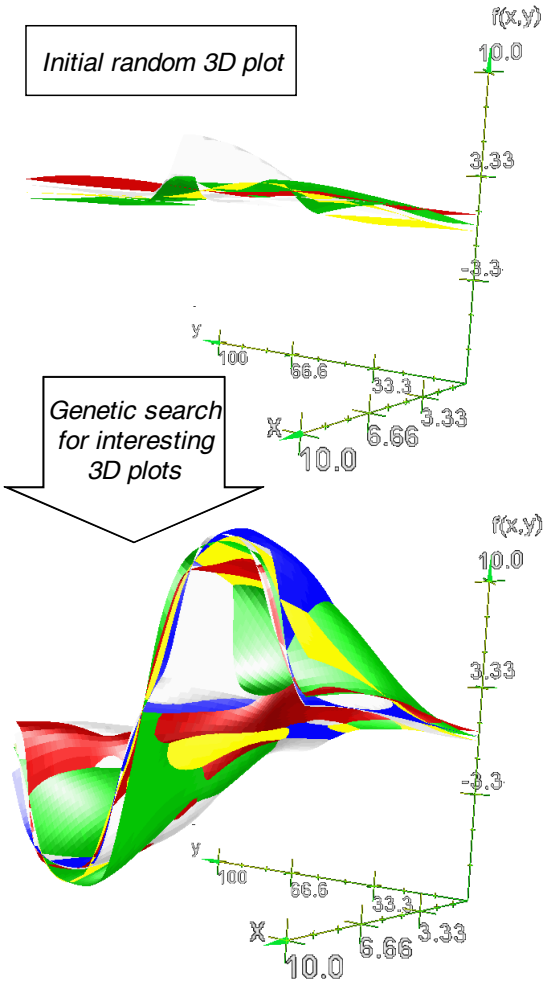

Figure 8: The initial plot is not interesting because it does not show any interesting behavior. Then the genetic algorithm is used to locate a more interesting plot in the multidimensional input space. The resulting 3d plot has much higher fitness - it defines the relationship among two input variables and the output variable very clearly. Also the credibility of the models is very high, as soon as their outputs correspond.

4 CONCLUSIONS

Visualization techniques presented in this paper were primary designed for visual data mining. However, they showed their usefulness also in model validation.

These visualizations help us to (a) estimate the credibility of models, (b) consider the quality of models, (c) discover the relationships of variables and identify true decision boundaries of classes.

We have also proposed the evolutionary search with a special fitness function designed to locate the most interesting plots in multidimensional input space. This is

particulary important because real world data sets traditionally have a high number of inputs and most of them are not very relevant.

ACKNOWLEDGMENTS

We would like to thank to our collaborators Jan Saidl and Jiří Nožka who implemented the visualization modules and the genetic algorithm.

This research is partially supported by the grant Automated Knowledge Extraction (KJB201210701) of the Grant Agency of the Academy of Science of the Czech Republic, and research programs "Transdisciplinary Research in the Area of Biomedical Engineering II" (MSM6840770012) and "Research in the Area of the Prospective Information and Navigation Technologies" (6840770014) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

REFERENCES

Abdel-Aal, R. 2005. Improving electric load forecasts using network committees. *Electric Power Systems Research* (74): 83–94.

GAME. The fake game environment for the automatic knowledge extraction. <neuron.felk.cvut.cz/game>.

Goldberg, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.

Hansen, L., and P. Salamon. 1990. Neural network ensembles. *IEEE Trans. Pattern Anal. Machine Intelligence* 12 (10): 993–1001.

Jardins, M., and P. Rheingans. 1999. Visualization of high-dimensional model characteristics. In *NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management*, 6–8. New York, NY, USA: ACM Press.

Keim, D. A. 1997. Visual techniques for exploring databases. <infovis.uni-konstanz.de/members/keim/PS/KDD97.pdf>.

Kordík, P. 2006, September. *Fully automated knowledge extraction using group of adaptive models evolution*. Ph. D. thesis, Czech Technical University in Prague, FEE, Dep. of Comp. Sci. and Computers, FEE, CTU Prague, Czech Republic.

Kordík, P., J. Saidl, and M. Šnorek. 2006. Evolutionary search for interesting behavior of neural network ensembles. In *IEEE Congress on Evolutionary Computation [CD-ROM]. Los Alamitos: IEEE Computer Society*, Volume 1, 235–238.

M. Gevrey, I. Dimopoulos, S. L. 2003. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling 160 (2003)*:249–264.

Roger, D. F. 2001. *An introduction to nurbs with historical perspective*. Morgan Kaufmann Publishers.

Santos, S. R., and K. W. Brodlie. 2002. Visualizing and investigating multidimensional functions. In *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, 173–ff. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association.

Stocker, T. 2003. Long-term perspectives on the earth system looking from the past into the future. Pleanry Talk IGBP Open Science Conference, Banff.

Theron, R., and J. D. Paz. 2006. Visual sensitivity analysis for artificial neural networks. In *Lecture Notes in Computer Science. Intelligent Data Engineering and Automated Learning - IDEAL 2006*, Volume 4224, 191–198: Springer-Verlag.

Tzeng, F.-Y., and K.-L. Ma. 2005. Opening the black box - data driven visualization of neural network. *IEEE Visualization 2005* 0:49.

UCI. Uci machine learning repository. <www.ics.uci.edu/~mlearn/MLSummary.html>.

Zhou, Z.-H., J. Wu, and W. Tang. 2002. Ensembling neural networks: Many could be better than all. *Artificial Intelligence* 137:239–263.

AUTHOR BIOGRAPHIES

IVO KONDAPANENI is a PhD student at the Czech Technical University in Prague, Faculty of Electrical Engineering where he also received his master's degree in Computer Science. His primary research area is information visualization, and he is also interested in visualization and modeling of implicit surfaces. His email address is <kondai1@fel.cvut.cz>.

PAVEL KORDÍK works as an assistant professor and researcher at the Department of Computer Science and Engineering, FEE, Czech Technical University in Prague, where he obtained his master's and Ph.D. degree in 2003 and 2007, respectively. He is the co-author of more than 20 publications. He is coordinator of the Automated Knowledge Extraction research project and a member of the research team of Transdisciplinary Research in the Area of Biomedical Engineering II research programme. His research interests are data mining, knowledge extraction, inductive models, neural networks, evolutionary computing, optimization methods, nature inspired continuous optimization, visualization of black-box behaviour and ensemble techniques. His email address is <kordikp@fel.cvut.cz>.

PAVEL SLAVÍK is a Full Professor of Computer Science at Czech Technical University in Prague - Czech Republic. His professional interests cover user interfaces and computer graphics (especially scientific visualization). He works as a member of the Computer Graphics Group at Department of Computer Science and Engineering. The group is the largest group of its kind in the country. He is an author of several textbooks used at the university. He has also written several dozens of papers published at conferences round the world. He is member of Eurographics, ACM, ACM SIGGRAPH and ACM SIGCHI. Every year he is a member of several program committees of conferences concerning user interfaces or computer graphics. His email address is <slavik@fel.cvut.cz>.