ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Interpretable Ensembles of Local Models for Safety-Related Applications

Sebastian Nusser[1,2], Clemens Otte[1], and Werner Hauptmann[1]

1- Siemens AG – Corporate Technology, Learning Systems
Otto-Hahn-Ring 6, 81730 Munich, Germany

2- Otto-von-Guericke-University of Magdeburg – School of Computer Science
Universitätsplatz 2, 39106 Magdeburg, Germany

Abstract. This paper discusses a machine learning approach for binary classification problems which satisfies the specific requirements of safety-related applications. The approach is based on ensembles of local models. Each local model utilizes only a small subspace of the complete input space. This ensures the interpretability and verifiability of the local models, which is a crucial prerequisite for applications in safety-related domains. A feature construction method based on a multi-layer perceptron architecture is proposed to overcome limitations of the local modeling strategy, while keeping the global model interpretable.

1 Introduction

Safety-related systems are systems whose malfunction or failure may lead to death or serious injury of people, loss or severe damage of equipment, or environmental harm. They are deployed, for instance, in aviation, automotive industry, medical systems and process control. This contribution discusses a machine learning approach for use in safety-related problems, an application domain where a wrong decision cannot be rectified. A more detailed discussion of this approach and its successful application to a real-world problem with high safety-requirements can be found in [1]. Alternative approaches for handling safety-related problems with machine learning methods are reviewed in [2].

In practical application tasks, the available training data is often too sparse and the number of input dimensions is too large to sufficiently apply statistical risk estimation methods. In most cases, high-dimensional models are needed to solve a given problem. Unfortunately, such high-dimensional models are hard to verify (*curse of dimensionality*), may tend to overfitting, and the interpolation and extrapolation behavior is often unclear or intransparent. An example of such counterintuitive and unintended behavior is illustrated in Fig. 1, where the prediction of the model changes in a region not covered by the given data set. Such behavior becomes even more likely and much more difficult to discover in the high-dimensional case. Thus, a model building method is required which provides a well-defined interpolation and extrapolation behavior[1].

The crucial aspect is to find a suitable trade-off between the generation of an interpretable and verifiable model and the attainment of a high predictive

[1] "Well-defined" states here that the decisions of the learned models can exactly be determined for every point of the input space.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

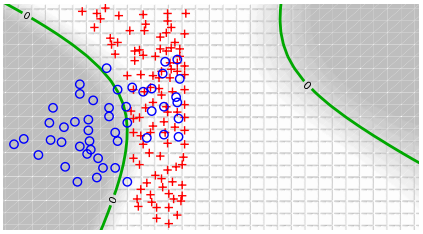


Fig. 1: Counterintuitive extrapolation behavior in a region not covered by the data set. This two-class problem is solved by a support vector machine (SVM) with an acceptable classification performance on the given data. However, in a region not covered by any data the decision of the SVM changes arbitrarily.

accuracy. It is obvious that complex models will be able to achieve a better performance on the available data. However, a higher complexity will lead to an increased effort for model verification.

The paper is organized as follows: Sect. 2 describes the local modeling strategy for safety-related domains. In Sect. 3 a feature construction method based on a multi-layer-perceptron architecture is presented to overcome limitations of the local modeling approach. An illustrative example is discussed in Sect. 4 and Sect. 5 concludes.

2 Local Modeling

This section discusses an approach utilizing the advantages of local modeling to deal with safety-related problems. The classification method, which was introduced in [1], is motivated by Generalized Additive Models [3, 4], and Separate-and-Conquer approaches [5]. This approach is designed to find an estimate of the unknown function $f : V^n \rightarrow Y$, where $V^n = \prod_{\imath=1}^{n} X_\imath$ with $X_\imath \subseteq \mathbb{R}$ and $Y = \{0, 1\}$, given an observed data set: $\mathfrak{D} = \{(\vec{v}_1, y_1), ..., (\vec{v}_m, y_m)\} \subset V^n \times Y$.

Basic Idea. The method introduced in the following is a greedy approach to find an additive estimate of the unknown function $f : V^n \rightarrow \{0, 1\}$. It is based on the projection of the high-dimensional data on low-dimensional subspaces. Local models are trained on these subspaces. By regarding only low-dimensional subspaces a visual interpretation becomes feasible and, thus the avoidance of unintended extrapolation behavior is possible. The ensemble of local models boosts the overall predictive accuracy and overcomes the limited predictive performance of each single local model, while the global model remains interpretable.

Projection of High-Dimensional Data. The projection π maps the n-dimensional input space V^n to an arbitrary subspace of V^n. This mapping is determined by a given index set $\beta \subset \{1, ..., n\}$. The index set defines the dimensions of V^n that will be included in the subspace V_β. Thus, the projection π on the input space V^n given the index set β is defined as: $\pi_\beta(V^n) = V_\beta = \prod_{\imath \in \beta} X_\imath$.

Local models. The j-th local model is defined as: $g_j : \pi_{\beta_j}(V^n) \rightarrow \{0, 1\}$, where β_j denotes the index set of the subspace where the classification error of the local model g_j is minimal. The final function estimate $\hat{f}$ of the global model

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Algorithm 1 Building an ensemble of local models.

parameter: $\mathfrak{D}$ – data set; c_{pref} – label of preferred class; dim_limit – limit of dimensions (fixed)

function models := build_model($\mathfrak{D}$, c_{pref})
  solve $\forall(\vec{v}, y) \in \mathfrak{D} : \min\{|y - g(\pi_\beta(\vec{v}))|\}$, $\beta \subset \{1, ..., n\}$ s.t.
    $|\beta| = \text{dim\_limit}$ and $\forall y = c_{\text{pref}} : |y - g(\pi_\beta(\vec{v}))| = 0$
  $\mathfrak{D}_{new} := \{(\vec{v}, y) | g(\pi_\beta(\vec{v})) = c_{\text{pref}}\}$
  if $(\mathfrak{D} \setminus \mathfrak{D}_{new} \neq \varnothing)$
    models := $\{g(\pi_\beta(\cdot))\} \cup$ build_model($\mathfrak{D}_{new}$, c_{pref})
  else
    models := $\varnothing$
  fi

Algorithm 2 Classifying new samples with an ensemble of local models.

parameter: $\vec{v} \in V^n$ – new sample data point; models – set of models returned by Algorithm 1

function class := evaluate_model($\vec{v}$, models)
  class := $\max_{g_j \in \text{models}} g_j(\pi_{\beta_j}(\vec{v}))$

is determined by the aggregation of the results of all local models $g_j(\pi_{\beta_j}(\vec{v}))$. The summation of the original Generalized Additive Model is replaced by an appropriate aggregation function, e.g. the max-operator of Algorithm 2.

Ensemble of Local Models. This method incorporates prior knowledge about the subgroups of the given problem and avoids hierarchical dependencies of the local models. It is required that the so-called *preferred class* c_{pref} must not be misclassified by any of the trained local models. This requirement typically leads to imbalanced misclassification costs. The local models are trained on low-dimensional projections of the high-dimensional input space with the goal to avoid the misclassification of the preferred class. A wrapper method for feature selection is used to determine the best projections. The local models greedily separate the samples of the other class from the preferred class samples. Missed samples of the other class are used to build further sub-experts. The algorithms for building such an ensemble of local models and for evaluating a new sample $\vec{v} \in V^n$ are shown in Algorithm 1 and Algorithm 2, respectively.

3 MLP-based Feature Construction

As demonstrated in [1], the ensemble of local models shows a good performance on real-world applications but there are problems that cannot be solved with the restriction to two- or three-dimensional local models. To overcome this limitation, a feature construction method based on a multi-layer perceptron (MLP) architecture is developed that generates low-dimensional linear combinations. The additionally generated input dimensions can be interpreted as preceding soft classifiers. The original input dimensions $V^n = X_1 \times X_2 \times ... \times X_n$ are used in the input layer and the target variable Y is used in the output layer. The hidden layer of this network consists of $n(n-1)/2$ nodes $hidden_{(i,j)}$, where $i, j \in \{1, ..., n\}$ and $i < j$. Each hidden node is only connected to two of the original input dimen-

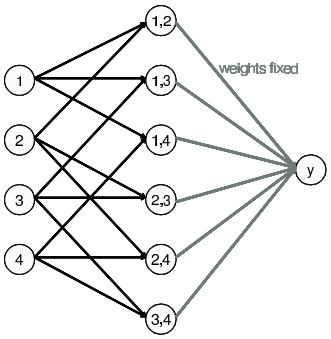


Fig. 2: MLP for feature construction.

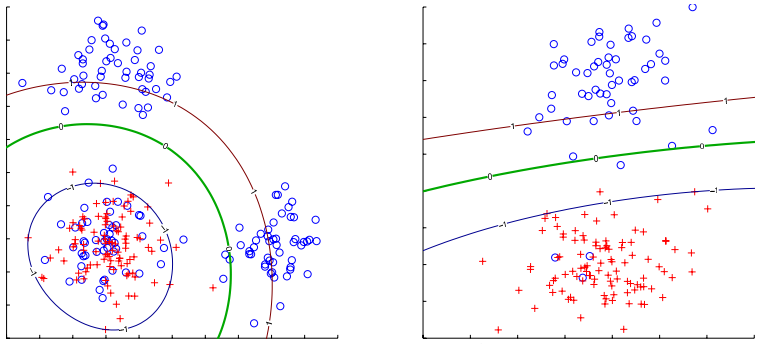
sions. The connections from the hidden to the output layer are set to 1 and are fixed during the network training procedure. This MLP architecture is depicted in Fig. 2. Due to this design, the network is forced in the hidden layer to find local classifiers on the given input dimension. The resulting weights of the hidden neurons can be used to build additional input dimensions. An additional input dimension is generated by: $X^{\text{new}}_{(i,j)} = \tanh(X_i \cdot w^{\text{hidden}}_{(i,j)} + X_j \cdot w^{\text{hidden}}_{(j,i)} + b^{\text{hidden}}_{(i,j)})$, where X_i, X_j are original input dimensions, $w^{\text{hidden}}_{(i,j)}$ is the connecting weight of input dimension X_i to hidden neuron $hidden_{(i,j)}$, and $b^{\text{hidden}}_{(i,j)}$ is the bias of the hidden neuron $hidden_{(i,j)}$. The additional input dimension $X^{\text{new}}_{(i,j)}$ can be seen as a preceding soft classifier.

Using all $n(n-1)/2$ additional input dimensions drastically increases the effort to determine the best projection of the data set. Thus, it is necessary to reduce the number of additional input dimensions by choosing only the "best" hidden neurons as additional input dimensions for the ensemble learning method. For instance, such selection can be performed by choosing the hidden neurons which are most correlated with the target variable.

4 An Illustrative Example

The Cubes data set is generated from four Gaussian components in a three-dimensional space. For each Class 1 cluster 50 samples are drawn from $N(e_i, 0.2 \cdot \mathbf{I})$, where e_i is a unit vector and $\mathbf{I}$ is the identity matrix. 100 samples of the Class 0 cluster are scattered around the origin, drawn from $N((0,0,0)^T, 0.2 \cdot \mathbf{I})$. All local models are trained as support vector machines (SVMs) with Gaussian kernel and the parameter set $\gamma = 0.2$ and $C = 5$.

Ensemble of Local Models. Class 0 is selected as the preferred class, $c_{\text{pref}} = 0$, i.e. this class must not be misclassified by any of the learned local models. In the example, this can be achieved by using imbalanced misclassification costs for Class 1 and Class 0, where the misclassification penalty of Class 0 is ten times higher than for Class 1. At the initial state, all two-dimensional projections of the Cubes data set are very similar. The best local model g_1, see Fig. 3(a), uses the projection $\pi_{\beta_1}(\vec{v})$ with $\beta_1 = \{1, 2\}$. 53 data points from Class 1 are misclassified by this local model. Thus, in the next iteration new



(a) 1st local model g_1 ($\beta_1 = \{1, 2\}$). (b) 2nd local model g_2 ($\beta_2 = \{2, 3\}$).

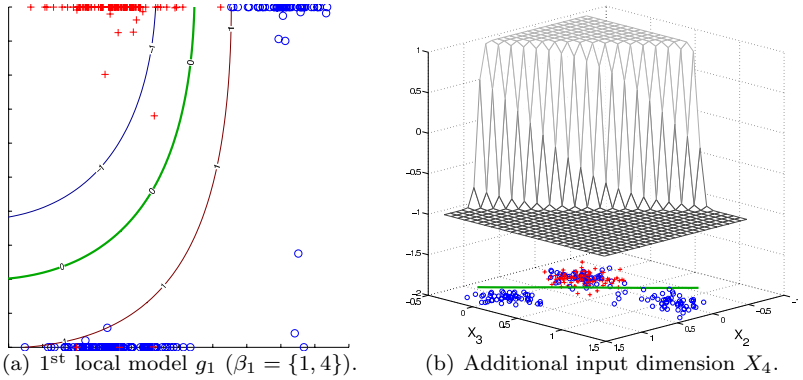
Fig. 3: The ensemble of local models and the CUBES data set: CLASS 1 samples are marked with blue circles and CLASS 0 samples are marked with red crosses. The decision boundary is labeled with zero and the margin of the SVM model is labeled with -1 and 1.

local models are trained only on samples, which are predicted as CLASS 0 by the first local model: $\mathfrak{D}_{new} = \{(\vec{v}, y) | g_1(\pi_{\beta_1}(\vec{v})) = 0\}$. In Fig. 3(b) the projection $\pi_{\beta_2}(\vec{v})$ with $\beta_2 = \{2, 3\}$ of the data set $\mathfrak{D}_{new}$ and the corresponding local model g_2 are shown. This local model misclassifies four CLASS 1 samples. Further improvements with the given parameter set are not possible. The final additive model is $\hat{f}(\vec{v}) = g_1(\pi_{\beta_1}(\vec{v})) \vee g_2(\pi_{\beta_2}(\vec{v}))$. Avoiding the misclassification of the preferred class $c_{\text{pref}} = 0$ leads to four misclassified CLASS 1 samples.

Feature Construction. By choosing CLASS 1 as preferred class, $c_{\text{pref}} = 1$, it becomes infeasible to solve the CUBES problem by an ensemble of local models with the restriction to two-dimensional submodels. In this case, the samples of the preferred class have a strong overlap with the samples of the other class in all two-dimensional projections. This problem can be solved by applying the feature construction method described in Sect. 3. Thus, it becomes possible to solve the CUBES problem with $c_{\text{pref}} = 1$ by a single two-dimensional local model. This local model is depicted in Fig. 4(a). It incorporates the original input dimension X_1 and the additionally generated input dimension $X_4 = X^{\text{new}}_{(2,3)}$, which is depicted in Fig. 4(b). This additional input dimension can be seen as preceding soft classifier that separates most of the CLASS 1 samples from the CLASS 0 samples. The resulting local model misclassifies only four CLASS 0 samples – further improvements are not possible. The performance of this solution is similar to the ensemble of local models with CLASS 0 as preferred class.

5 Conclusions

To be able to apply machine learning approaches in the field of safety-related problems it is crucial to provide interpretable and verifiable models. Since it is infeasible to sufficiently interpret high-dimensional models, such complex models are not applied to safety-related applications. On the other hand, simple models, which are easier to interpret, show a lack of predictive performance.



(a) 1st local model g_1 ($\beta_1 = \{1, 4\}$). (b) Additional input dimension X_4.

Fig. 4: Feature construction for use with the ensemble of local models and the CUBES data set: CLASS 1 samples are marked with blue circles and CLASS 0 samples are marked with red crosses. The decision boundary is labeled with zero and the margin of the SVM model is labeled with -1 and 1.

The binary classification approach discussed in this paper provides a good trade-off between the interpretation and verification of the learned (local) models, avoiding an unintended extrapolation behavior, and the achievement of a high predictive accuracy. Each local model can be interpreted visually and the ensemble of the local models compensates for the limited predictive performance of each single local model. The local models can be evaluated by a domain expert to avoid unintended extrapolation and interpolation behavior. In contrast to dimensionality reduction methods, which combine several dimensions of the input space, the local models are trained on the original dimensions, allowing the experts to directly evaluate the trained models. For problems that cannot be solved with the restriction to low-dimensional local models, the MLP-based feature construction method can compensate for this limitation of the local modeling procedure while the models remain interpretable. The additionally generated input dimensions can be regarded as preceding soft classifiers. The described approach has successfully been deployed in a safety-related application in the area of automotive safety electronics [1]. Currently, the introduced classification approach is being extended to also solve multi-class problems.

References

[1] Sebastian Nusser, Clemens Otte, and Werner Hauptmann. Learning binary classifiers for applications in safety-related domains. In *Proceedings of 17th Workshop Computational Intelligence*, pages 139–151. Universitätsverlag Karlsruhe, 2007.

[2] Clemens Otte, Sebastian Nusser, and Werner Hauptmann. Machine learning methods for safety-related domains: Status and perspectives. In *Proceedings of the Symposium on Fuzzy Systems in Computer Science*, pages 139–148, Magdeburg, Germany, 2006.

[3] Charles J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13(2):689–705, 1985.

[4] Trevor Hastie and Robert Tibshirani. *Generalized Additive Models*. Chapman&Hall, 1990.

[5] Johannes Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.