ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Heterogeneous mixture-of-experts for fusion of locally valid knowledge-based submodels

Jörg Beyer[1,2], Kai Heesche[1], Werner Hauptmann[1], and Clemens Otte[1]

1- Siemens AG - Corporate Technology, Information and Communications, Learning Systems
Otto-Hahn-Ring 6, 81739 Munich, Germany

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

Abstract. Real-world applications often require the joint use of data-driven and knowledge-based models. While data-driven models are learned from available process data, knowledge-based models are able to provide additional information not contained in the data. In this contribution, we propose a method to divide the input space on the basis of the validity ranges of the knowledge-based models. By doing so they are only active in those domains they are designed for. The data-driven models complete the coverage of the input space. We demonstrate the benefits of our approach on a real-world application for the energy management of a hybrid electric vehicle.

1 Introduction

To generate adequate models for complex real-world applications the consideration of available knowledge-based and data-driven models is required. The integration of knowledge-based models is of particular interest since they are able to provide information not contained in the training data, for example, process knowledge from human domain experts, which may be given in different forms like rules, look-up tables, or physical equations. As these models are only locally valid it is crucial to include their validity ranges in the training process.

There are different approaches to learn local models, like radial basis function networks [1], boosting [2], switching regression [3], [4], or mixture-of-experts models [5], [6]. The algorithms for learning local models can be discriminated with respect to several aspects: in the way they partition the input space into different regions, how they divide the training data into subsets, the type of submodels they use, or how they accumulate the outputs of the submodels. However, these approaches are developed to learn from scratch, i.e. they train their local models only with available training data.

The objective of the proposed method, referred to as heterogeneous mixture-of-experts (HME), is to integrate different kinds of knowledge-based models and to use data-driven models to close remaining gaps between different knowledge-based models with respect to the input space coverage. The knowledge-based models are designed for particular regions of the input space and are thus only valid in those regions. In order to ensure that the models are only active in

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

regions they are designed for their specific validity ranges have to be included in the process of dividing the input space.

The paper is organized as follows: Sect. 2 describes the approach for integrating information about the validity ranges of the knowledge-based models. In Sect. 3, some experiments on a real-world application are outlined. The results of the model are compared with a conventional mixture-of-experts (ME) model, a radial basis function (RBF) network, and a multilayer perceptron (MLP). Sect. 4 concludes the paper.

2 HME Model

The conventional ME model consists of a set of submodels that perform a local function approximation [5], [6]. It learns to decompose complex problems into simpler, easier to solve subproblems. This decomposition is learned by a gate function which partitions the input space and assigns submodels to these regions. In contrast to the ME model, the introduced HME model starts with some predefined submodels, representing different subprocesses of the application. To ensure that the predefined submodels are assigned to those domains of the input space they are designed for, information about the specific validity ranges of the predefined knowledge-based submodels is used for the partitioning of the input space.

From the probabilistic perspective the output of the HME model can be interpreted as the probability of generating output $y^{(n)}$ given input vector $\vec{x}^{(n)}$:

$$P\left(y^{(n)} \left| \vec{x}^{(n)}, \Theta\right.\right) = \sum_{j=1}^{M} g_j\left(\vec{x}^{(n)}, \theta_g\right) P\left(y^{(n)} \left| \vec{x}^{(n)}, \theta_j\right.\right),$$

where M is the number of submodels, Θ is the set of parameters $\left\{\theta_g, \{\theta_j\}_{j=1}^{M}\right\}$ of the gate and the submodels. The mixture coefficient $g_j\left(\vec{x}^{(n)}, \theta_g\right)$ of submodel j is constrained to $g_j\left(\vec{x}^{(n)}, \theta_g\right) \geq 0$, $\forall j = 1, \ldots, M$ and $\sum_{j=1}^{M} g_j\left(\vec{x}^{(n)}, \theta_g\right) = 1$. The probability $P\left(y^{(n)} \left| \vec{x}^{(n)}, \theta_j\right.\right)$ represents the conditional densities of target $y^{(n)}$ for model j.

The training process is similar to that of the ME model. For each knowledge-based submodel j a mapping $v_j : \Re^n \rightarrow [0, 1]$, $\forall j = 1, \ldots, M$ is given that computes a validity value of the submodel for an input vector. The specific validity function of a knowledge-based submodel j for the i-th dimension is

$$v_j\left(x_i^{(n)}\right) = \left(\frac{1}{1 + \exp\left(s_j\left(x_i^{(n)} - u_{ji}\right)\right)} - \frac{1}{1 + \exp\left(s_j\left(x_i^{(n)} - l_{ji}\right)\right)}\right),$$

where l_{ji} and u_{ji} are the lower and the upper bound of the validity range of submodel j in dimension i. The parameter s_j determines the slope of the border of the validity range. An example of the behavior of a validity function is

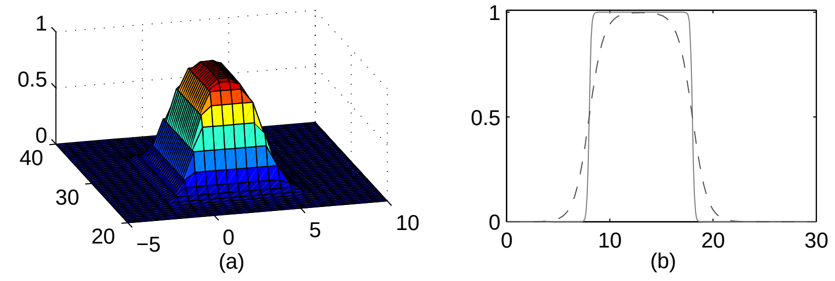

Fig. 1: (a): validity range of a model in a two-dimensional input space, (b): slope of the border depends on s for s=1 (dashed line) and s=8 (solid line).

depicted in Fig. 1. Fig. 1 (a) shows the validity range in a two-dimensional input space. The influence for different values of s_j is sketched in Fig. 1 (b). For small s_j the slope of the border is more flat. The higher s_j gets, the steeper is the slope of the border. In this way the transition between the submodels can be controlled. If there are smoothness assumptions about the target function one can choose a lower value for s_j.

To update the model parameter the EM algorithm is used. In the expectation step, the validity values are integrated into the computation of the posterior probability $h_j^{(n)}$ of selecting submodel j for input vector $\vec{x}^{(n)}$:

$$h_j^{(n)} = \frac{v_j\left(\vec{x}^{(n)}\right) g_j\left(\vec{x}^{(n)}, \theta_g\right) P\left(y^{(n)} \left| \vec{x}^{(n)}, \theta_j\right.\right)}{\sum_{k=1}^{M} v_k\left(\vec{x}^{(n)}\right) g_k\left(\vec{x}^{(n)}, \theta_g\right) P\left(y^{(n)} \left| \vec{x}^{(n)}, \theta_k\right.\right)}. \tag{1}$$

This enforces the gate to reduce the weights of submodel outputs if the input vectors are located outside their domains. The particular amount of weight decrease depends on the value of v_j.

In the maximization step, the error function E

$$E = \sum_{n=1}^{N}\sum_{j=1}^{M} h_j^{(n)} \log\left(g_j\left(\vec{x}^{(n)}, \theta_g\right) P\left(y^{(n)} \left| \vec{x}^{(n)}, \theta_j\right.\right)\right),$$

decomposes into separate maximization problems for the gate

$$E_g = \sum_{n=1}^{N}\sum_{j=1}^{M} h_j^{(n)} \log g_j\left(\vec{x}^{(n)}, \theta_g\right), \tag{2}$$

and for each data-driven submodel j

$$E_j = \sum_{n=1}^{N} h_j^{(n)} \log P\left(y^{(n)} \left| \vec{x}^{(n)}, \theta_j\right.\right). \tag{3}$$

The different data-driven models are learned with corresponding standard opti-

Algorithm 1 HME-Algorithm

```
input: data set D, knowledge-based models, validity ranges, number of
data-driven models
output: HME model
repeat
    compute $h_j^{(n)}$, cf. Eq. (1)
    learn gate, cf. Eq. (2)
    learn data-driven submodels, cf. Eq. (3)
until termination criterion fulfilled
```

mization algorithms, e.g. gradient descent for a neural network. The algorithm for building an HME model is summarized in Algorithm 1.

The following section describes some experiments for a real-world application. It is shown that the HME is able to integrate different knowledge-based models and achieve a superior performance.

3 Simulation of Energy Flow in a Hybrid Electric Vehicle

In this application, the energy flow in the powertrain of a hybrid electric vehicle is simulated. Charge and discharge energy flows of the battery can be distinguished. By applying the available expert knowledge, four distinct driving modes can be defined: pure electric drive mode, hybrid drive mode, brake mode, and drag mode. In pure electric drive mode and hybrid drive mode energy is provided by the battery to drive the electric motor. In brake mode and drag mode the electric motor is operating as a generator to regenerate the kinetic energy used for charging the battery. Furthermore, the battery must maintain certain chemical limits. These limits determine the maximum charge and discharge capabilities of the battery dependent on its state of charge and temperature. Using knowledge from domain experts, specific models for each mode were designed.

In this application, the hybrid electric vehicle data set includes five input features and about 400.000 samples. The data set is randomly divided into a training data set (80% of the data) and a test data set (20% of the data). We used the mean absolute error to estimate the predictive error:

$$\hat{e} = \frac{1}{N}\sum_{n=1}^{N}\left|y^{(n)} - f\left(\vec{x}^{(n)}\right)\right|.$$

The overall experiment is performed ten times and the results are averaged. The following models were compared: an HME, an ME, an RBF network, and an MLP. The HME model uses four expert models. Two characteristic maps and a mathematical model represent the pure electric drive mode, brake, and drag mode. However, since the hybrid drive mode is too complex to provide an adequate mathematical model, for this mode a two-layer MLP with five input units, four hidden units and one output unit was learned. Each mode has different input features. As gate, an MLP with four hidden units was applied.

Model	Predictive error (std)	
	training	testing
HME	1.95 (0.018)	2.00 (0.021)
ME	3.15 (0.041)	3.20 (0.044)
RBF	5.31 (0.059)	5.37 (0.058)
MLP	3.08 (0.045)	3.12 (0.056)

Table 1: Predictive error for the hybrid electric vehicle data set.

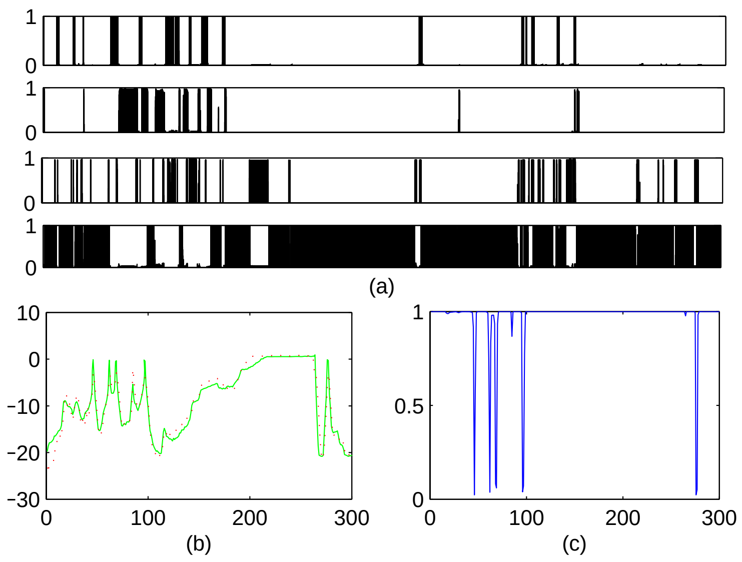

Fig. 2: (a): gate outputs for a data sequence for the different submodels, (b): weighted output of the brake model (green solid line) for data of the corresponding mode and the corresponding target values (red dotted line) for the energy flows, (c) gate outputs for the brake model for the data from (b).

The ME consists of four MLPs with four hidden units and as gate an MLP with four hidden units was used. The RBF network uses six Gaussian basis functions and the single MLP comprises eight hidden units.

The results for the hybrid vehicle data are shown in Table 1. The HME has achieved superior performance due to the incorporation of knowledge-based models. Fig. 2 (a) shows the mixture coefficients of the submodels of the HME model. In most cases, the gate selects only one submodel for each input vector. This behavior is consistent with the knowledge of the domain expert that the submodels were defined for different mutually exclusive modes. Fig. 2 (b) and (c) illustrates the assignment of data of the brake mode to the corresponding brake submodel by the gate. Fig. 2 (b) depicts the weighted contribution of the brake model for data of its domain. The corresponding mixture coefficients are shown in Fig. 2 (c). The brake submodel is exclusively responsible for data in its domain. We conclude that the gate has learned the correct partitioning of the input space. Each submodel is only responsible for data from its mode. In contrast to the HME, the ME model was not able to distinguish those modes and partitioned the input space in a technically non-plausible way.

Furthermore, except the HME model, all other models violated chemical limitations because they predicted energy flows that cannot be provided by the battery. The required information about these limits is not contained in the data, but in the knowledge-based models.

4 Conclusions

Applying the proposed HME model, it is possible to use existing predefined models and to complete these by data-driven submodels. To be able to integrate given knowledge-based models into the process of simultaneously training the data-driven submodels and a gate model it is crucial to incorporate the validity ranges of the knowledge-based models. The integration of knowledge-based models does not only lead to a superior performance but also results in an improved plausibility and reliability of the HME model compared to the other models. In addition, the HME benefits from the knowledge-based models because these also carry information not contained in the data as shown in the application example.

Future work considers the problem of sparse data or regions of the input space not covered by training data. The behavior of the model prediction becomes counterintuitive in such regions of the input space which are not covered by the data set. Thus, we predefine the gate in regions where no training data is available. In this way, extrapolation of the gate in such regions of the input space can be avoided. Furthermore, we investigate the derivation of a confidence measure based on the validity ranges.

References

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation, 2nd edition*, NJ: Prentice Hall, Upper Saddle River, 1999.

[2] Y. Freund and R. E. Schapire, Decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, 55(1):119-139, 1997.

[3] R. E. Quandt, A new approach to estimating switching regressions, *Journal of the American Statistical Association 67*, 306-310, 1972.

[4] K. N. Lau, C. H. Yang, and G. V. Post, Stochastic preference modeling within a switching regression framework, *Computers & Operations Research*, 23(12):1163-1169, 1996.

[5] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, Adaptive mixtures of local experts. *Neural Computation*, 3:79-87, 1991.

[6] M. I. Jordan and R. A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, *Neural Computation*, 6(2):181-214, 1994.