Annealed Competition of Experts for a Segmentation and Classification of Switching Dynamics

Klaus Pawelzik ‡*, Jens Kohlmorgen † , Klaus-Robert Müller +†

‡ Institut für Theoretische Physik and SFB 185 Nichtlineare Dynamik
Universität Frankfurt, 60054 Frankfurt/M., Germany
† GMD–FIRST (German National Research Center for Computer Science)
Rudower Chaussee 5, 12489 Berlin, Germany
+ Department of Mathematical Engineering and Information Physics
University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113, Japan

Abstract

We present a method for the unsupervised segmentation of data streams originating from different unknown sources which alternate in time. We use an architecture consisting of competing neural networks. Memory is included in order to resolve ambiguities of input-output relations. In order to obtain maximal specialization, the competition is adiabatically increased during training. Our method achieves almost perfect identification and segmentation in the case of switching chaotic dynamics where input manifolds overlap and input-output relations are ambiguous. Only a small dataset is needed for the training proceedure. Applications to time series from complex systems demonstrate the potential relevance of our approach for time series analysis and short-term prediction.

1 Introduction

Neural networks provide frameworks for the representation of relations present in data. Especially in the fields of classification and time series prediction, neural networks

*Corresponding author, email:klausp@salk.edu, temporary adress: The Salk Institute, CNL, Box 85800, San Diego, CA 92186-5800.

have made substantial contributions. An important prerequisite for the successful application of such systems, however, is a certain uniformity of the data.

In most analysis of data series, stationarity must be assumed, i.e. it must be assumed that the relations remain constant over time. If, on the contrary, the data originate from different sources, e.g. because the underlying system *switches* its dynamics, standard approaches like simple multi-layer perceptrons are likely to fail to represent the underlying input-output relations. Such time series can originate from many kinds of systems in physics, biology and engineering. Phenomena of this kind include e.g. speech (Rabiner 1988), brain data (Pawelzik 1994), and dynamical systems which switch their attractors (Kaneko 1989).

In this paper we present a method for the segmentation of such data streams without prior knowledge about the sources. We consider the case where the different input-output samples $(x(t), y(t))$ are generated by a number n of unknown functions $f_{l(t)}, l = 1, ..., n$ which alternate according to $l(t)$, i.e. $y(t) = f_{l(t)}(x(t))$. The task then is to determine both, the functions f_l together with their respective attributions $l(t)$ from a given time series $\{(x(t), y(t))\}_{t=0}^{N}$. Since both the functions and the segmentation are considered to be unknown, they have to be determined simultaneously, i.e. the correct segmentation has to be found in an unsupervised manner.

The mixtures of experts architecture, as proposed by Jacobs et al. (1991), potentially offers a solution to this problem, since it can represent different functions by the respective experts. There are, however, problems when applying the mixture of experts architecture to the task of identifying alternating sources.

One problem arises, when the gating of the experts is based on the input alone, because in general the underlying sources will have overlapping input domains. In order to solve this problem, we here use an ensemble of expert-networks whose competition depends only on their relative performance and *not* on the input. This way of introducing the competition relates to clustering and vector quantization (McLachlan and Basford 1988) and is in contrast to the mixtures of experts architecture that uses an input-dependent gating-network (Jacobs et al. 1991).

When the sources have overlapping arguments, a further problem arises: the functions may intersect. In this case, there are input-output pairs which are identical for different functions $i \neq j$, i.e. there are (x, y) for which $y = f_i(x) = f_j(x)$. As we will show, such intersections induce additional ambiguities, a further problem, which can only be resolved by imposing additional constraints. We present a learning rule performing this disambiguation, which is derived from a simple assumption about memory in the switching process: a low switching rate. This assumption allows one to train the system of experts on very small data sets and does not require *any* statistics of switching events. In particular the method can identify switchings in a time series from only a number of data which just suffices to characterize the two respective functions.

Our approach does not provide an analysis of the dynamics of the switching itself, which has been adressed in (Cacciatore and Nowlan 1994) and (Bengio and Frasconi 1994) and we discuss the relation of these approaches to our work in section 5.

For unique segmentation, each sample $(x(t), y(t))$ must be assigned to only one expert. This can most easily be achieved by considering only the respective best performing expert. However, when using such hard competition during training, it is likely to get stuck in local minima, which in simple cases can be overcome by using sample dependent ad hoc initializations (Kohlmorgen et al. 1994, Müller et al. 1994, Müller et al. 1995). As a more general approach, we here propose to anneal the competition of the networks *adiabatically* during training (see also Yuille et al., 1994). We will show that with this method the networks successively specialize in a hierarchical manner via a series of phase transitions, an effect which has been analysed in the context of clustering by Rose et al. (1990).

In section 2, we introduce our approach and in section 3 we demonstrate the features of our method with an example of alternating functions over the unit interval which intersect. In that example, the input-output samples are given by the dynamics of chaotic maps and the experts correspond to predictors. This relates our method to common techniques in system identification (Shamma and Athans, 1992), and time series prediction (Tong and Lim, 1980). In section 4, we apply our method to benchmark data from the Santa Fe Time Series Prediction Competition (Weigend and Gershenfeld 1994), an application which demonstrates that our approach may substantially improve predictions of time series and opens new perspectives for signal classification, which we finally discuss in section 5.

2 Unmixing of Experts

Data originating from different sources are subject to ambiguity. If input-output relations are considered, this can have at least two interdependent reasons. First, the input domains may overlap. However, it is impossible for a single network to map the same inputs to different outputs without using extra information. Second, input *and* output of different sources can be identical for a subset of the data. In this latter case, information beyond the input-output pairs is required in order to reassign the data to the sources.

For illustrating the basic ideas underlying our approach we discuss the extreme case of completely overlapping input manifolds. An example is given by input-output pairs $(x_t, y_t) = (x_t, f_l(x_t)), t = 1, \ldots, T$, that at each time step t are a choice $l = l(t), l = 1, 2, 3, 4$ of one of the four maps $f_1(x) = 4x(1-x), x \in [0,1]$ ("logistic map"), $f_2(x) = \{2x$ if $x \in [0, .5)$ and $2(1-x)$, if $x \in [.5, 1]\}$ ("tent map"), $f_3 = f_1 \circ f_1$ ("double logistic map") or $f_4 = f_2 \circ f_2$ ("double tent map"). $f \circ f$ denotes the iteration $f(f(x))$. If we set $x_{t+1} = y_t$, we get a chaotic time series $\{x_t\}$ with $x_{t+1} = f_l(x_t)$, see Fig.1. When these maps are alternately used, a given input x_t alone does not determine the appropriate output y_t, and a representation of the underlying relations therefore must *necessarily* contain a division into subtasks. For such data sets, a gating network that depends only on the input (Jacobs et al. 1991) must necessarily fail.

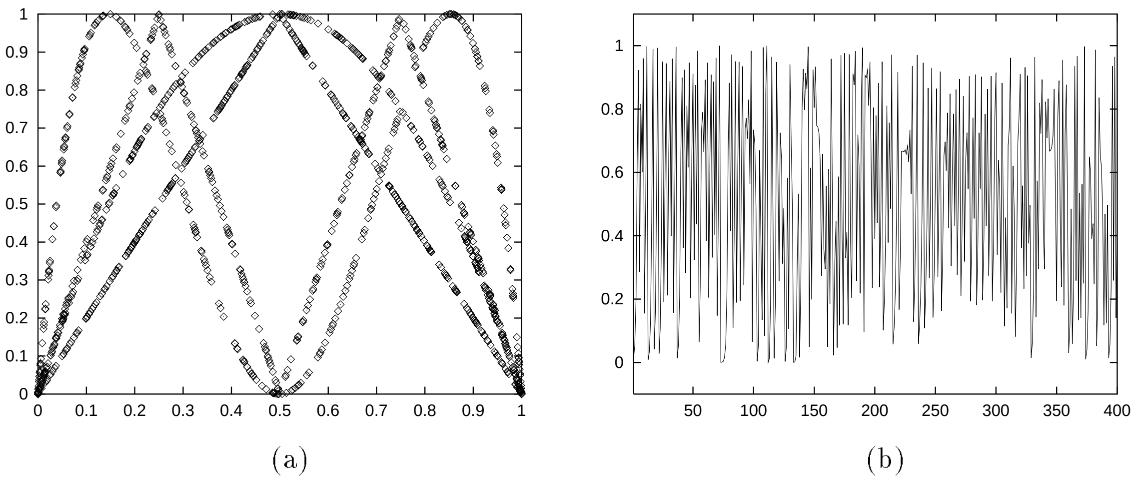


Figure 1: *(a) Training data drawn from four chaotic return maps, 300 points for each map. A new map is chosen after every 100 recursions. The first 400 values of the resulting time series are shown in (b).*

In our approach, we therefore adapt a set of predictors $\tilde{f}_i, i = 1, \ldots, n$, weighted only by their relative performance. The optimal choice of function approximators $\tilde{f}_i$ depends on the specific application. Throughout this paper we are using radial basis function networks (RBFN's) of the Moody-Darken type (Moody and Darken 1989), because they offer a fast learning method. We train the weights w_i of network i by performing a gradient descent

$$\Delta w_i \propto -\sum_t p_i^t \frac{\partial \varepsilon_i^t}{\partial w_i} \tag{1}$$

on the squared errors

$$\varepsilon_i^t = \left(\tilde{f}_i(x_t) - y_t\right)^2. \tag{2}$$

The weighting coefficient p_i^t corresponds to the relative probability for a contribution of network i and the p_i^t are constrained to be $\sum_i p_i = 1$. Our approach differs from previous work in the way the p_i^t's incorporate memory that is present in the switching process. We start by assuming that the outputs $\tilde{f}_i(x_t)$ are distributed according to Gaussians, i.e.

$$p(\varepsilon \mid i) \propto e^{-\beta\varepsilon_i}. \tag{3}$$

We furthermore assume that the system does not switch its state $l(t)$ every time step, but instead alternates among the different subsystems $i \neq j$ at low rates $r_{ij} < r$, which is a rather weak assumption about the memory of the switching process, which we will use in the following to derive a simple bias in the probabilities p_i^t.

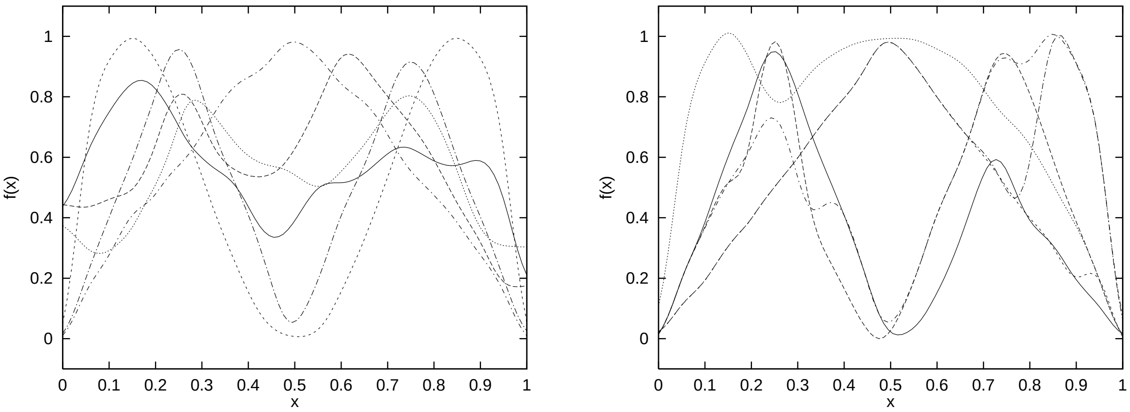
Then, the probability that a given subsequence $\sigma_t^\Delta = [(x(t-\Delta), y(t-\Delta)), \ldots, ((x(t+\Delta), y(t+\Delta))]$ of the time series is generated by a particular sequence $\vec{s}_t^\Delta = (l(t-\Delta), \ldots, l(t+\Delta))$ of functions $f_{l(t)}$ is given by

$$p(\sigma_t^\Delta \mid \vec{s}_t^\Delta) = p(\vec{\varepsilon}_t^\Delta \mid \vec{s}_t^\Delta) = \prod_{\tau=-\Delta}^{\Delta} p(\varepsilon_{l(t+\tau)} \mid l(t+\tau)), \tag{4}$$

where $\vec{\varepsilon}_t^\Delta$ denotes the corresponding sequence of errors. Bayes' rule in this case gives

$$p(\vec{s}_t^\Delta \mid \sigma_t^\Delta) = \frac{p(\vec{\varepsilon}_t^\Delta \mid \vec{s}_t^\Delta) p(\vec{s}_t^\Delta)}{\sum_{\vec{s'}^\Delta} p(\vec{\varepsilon}_t^\Delta \mid \vec{s'}^\Delta) p(\vec{s'}^\Delta)}, \tag{5}$$

where the sum runs over all possible sequences $\vec{s'}^\Delta$. This equation can be strongly simplified in case of a low bound r on the switching rate when short sequences have a small probability q to contain a switching event, i.e. if $\Delta \leq q/r$.



(a) (b)

Figure 2: *(a) Result for hard competition without prior annealing: Although a proper initialization was intended, one net grabbed two "similar" return maps, f_1 and f_2. A distinction between these two maps is no longer possible and the prediction error for both maps remains high. (b) Annealing without the inclusion of memory allows the creation of maps that jump from one target map to another along the x-axis. The information, that consecutive data points belong with high probability to the same dynamics, is not utilized.*

In this case, we can neglect sequences which contain switchings, i.e. we set $p(\vec{s}^{\Delta}) = 0$ if not all components are equal. The remaining n sequences are considered equiprobable according to maximum entropy (i.e. $p(\vec{s})^{\Delta} = 1/n$), and we then obtain from Eq.(4) and (5)

$$p(l \mid \vec{\varepsilon}_t^{\Delta}) = \frac{\prod_{\tau=-\Delta}^{\Delta} p(\varepsilon_l^{(t-\tau)} \mid l)}{\sum_{k=1}^{n} \prod_{\tau=-\Delta}^{\Delta} p(\varepsilon_k^{(t-\tau)} \mid k)}, \tag{6}$$

the probability that f_l generated the subsequence σ_t^{Δ}. Using Eq.(3), this finally provides the estimate for the weighting coefficients:

$$p_i^t = \frac{e^{-\beta \sum_{\tau=-\Delta}^{\Delta} \varepsilon_i^{t-\tau}}}{\sum_{j=0}^{n} e^{-\beta \sum_{\tau=-\Delta}^{\Delta} \varepsilon_j^{t-\tau}}}. \tag{7}$$

Note, that this result equivalently, but less intuitively, can be derived from maximizing the log likelihood of the observation $F = \sum_t log\{\sum_i p(\varepsilon_i^t \mid i)p(i)\}$ under the above assumptions. For $\Delta = 0$, this reduces to a mixtures of gaussians (McLachlan and Basford 1988) and is equivalent to a mixture of experts (Jacobs et al. 1991), however, *without* a gating network.

According to Eq.(7), we can simply use low-pass filtered errors instead of the plain ε_i^t in order to include memory that originates from a low switching rate. The drastic simplification of memory (probability 0 for sequences of length Δ which include switching) led to the box–type filter, which might be replaced by an exponential[1] in order to model the switching-probabilities more realistically. Yet, without any knowledge about the characteristics of the time-series, Eq.(7) seems to be the simplest and at the same time computationally least expensive way to include memory. Heuristically, Eq.(7) is analogous to evolutionary inertia, since once a predictor has performed better than its competitors, it also has an advantage for temporally adjacent data points. This helps to regularize data at ambiguities. In the example of the chaotic maps, such ambiguities emerge at the intersections, where additional information is required to decide which branches of the function "belong together".

For the purpose of segmentation, it might seem to be most desirable to choose β large. Indeed, one could consider $\beta = \infty$, which corresponds to hard competition (winner-takes-all) and guarantees an unambiguous segmentation (Kohlmorgen et al. 1994, Müller et al. 1994, Müller et al. 1995). We found, however, that using hard competition right from the beginning does not always lead to a sufficient diversification of the predictors. The final result in general depends on the choice of initial parameters which may lead to local minima in the likelihood F, and a mixing of maps can occur (see Fig. 2(a)).

We solve this initialization problem by *adiabatically* increasing the degree of competition. For $\beta = 0$, the predictors equally share the same data for training. Increasing β enforces the competition, thereby driving the predictors to a specialization on different subsets of the data. Diversification occurs at particular "temperatures" $T = 1/\beta$ and the network parameters separate abruptly, resolving the underlying structure to more detail. These phase transitions are indicated by a drop of the mean squared error $E = \sum_t \sum_i p_i^t \varepsilon_i^t$ (see Fig.3(a)) and have been described within a statistical mechanics formalism (Rose et al. 1990). Note, that a careful decrease of T is crucial when fine differences of underlying functions have to be resolved.

[1] The latter would yield a weighted low-pass filter in Eq.(7).

3 Applications to Switching Chaos

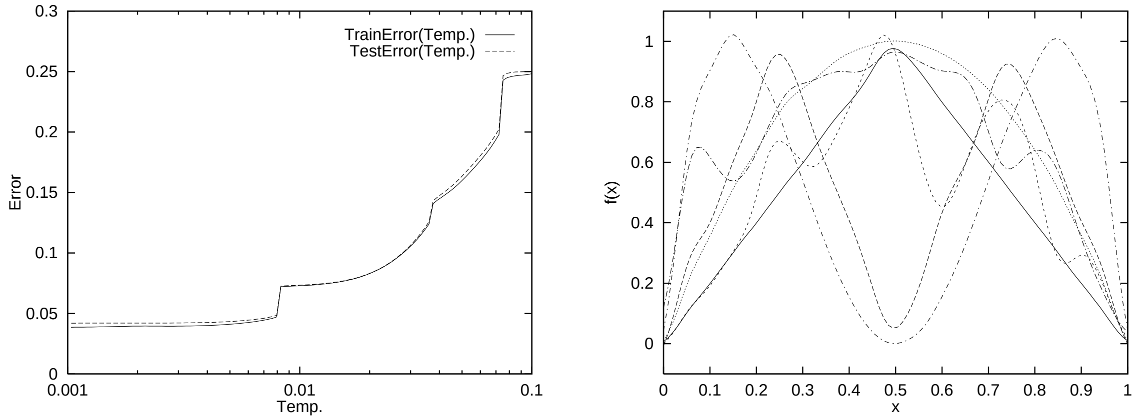
First we illustrate our approach with a time series of $N = 1200$ points from the four chaotic maps $f_1, ..., f_4$ introduced above (Fig.1). These maps were alternated every 100 iteration steps. Because these dynamical systems are ergodic on the support $x \in [0, 1]$, they cannot be distinguished on the basis of their arguments alone. Furthermore, the small rate $r_s = 1/100$ guarantees a large probability for short sequences of e.g. length $l = 7$ to contain no alternations of the underlying system, which justifies our simple method of taking memory into account by setting $\Delta = 3$ in Eq.(7). Note however, that this parameter is not crucial.

We used 6 radial basis function networks of the Moody-Darken type (Moody and Darken 1989) as predictors and decreased the temperature $T = 1/\beta$ adiabatically, i.e. the next smaller value of the temperature is taken, when the overall error E had saturated. The result is shown in Fig.3. The error decreases most during phase transitions (Fig.3(a)), which occur when the different underlying dynamics abruptly become resolved to more detail (Fig.4). After the relevant structures have been found by the algorithm, no further phase transitions occur and there is only little further decrease of the error when T approaches zero. At $T \simeq 0$, we find that four networks (out of six) segmented the time series almost exactly at the switching points, while two drifted off (Fig. 3b), did not contribute, and therefore could be removed without changing the performance E.

The method can be applied to time series from high-dimensional chaotic systems simply by replacing the scalar argument x by vectors which are obtained by the method of time delay embedding of the time series (Takens 1981, Liebert et al. 1991) and by a corresponding adaptation of the networks. As an example for a high-dimensional chaotic system, we take the Mackey-Glass delay-differential equation

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-t_d)}{1 + x(t-t_d)^{10}}, \tag{8}$$

originally introduced as a model of blood cell regulation (Mackey and Glass 1977). We generated a time series of $N = 400$ points where we switched the delay parameter t_d. For the first and last 100 samples (sampling rate $\tau = 6$) we chose $t_d = 17$, whereas for the second 100 samples we used $t_d = 23$ and for the third $t_d = 30$. To increase the difficulty of the problem, 5% noise was added at each integration step, thereby turning the system stochastic (Fig.5(a)). For the creation of a training set out of this time series, an embedding dimension $m = 6$ was used (Casdagli 1989).



(a) (b)

Figure 3: *(a) Training and test error during the annealing process both indicate phase transitions. (b) The maps learned by the RBFN's at the end of the process. Four nets have specialized on each of the given dynamics, while two nets dropped off and finally did not contribute to the segmentation and the overall error E.*

During training, two phase transitions occurred (Fig.5(b)), indicating that the system detected the different dynamical systems. The second transition (at $T \approx 0.0007$) becomes more prominent when simpler networks are used. However, this leads to sub–optimal prediction results and was therefore not applied. The removal of three nets at $T \simeq 0$ did not increase the error significantly (Fig.5(c)), which correctly indicates that three predictors completely describe the source. Segmentation, finally, was perfect (Fig.5(d)). The performance (convergence speed, segmentation accuracy) of our approach with the high-dimensional Mackey-Glass data was even better than for the one-dimensional maps, which indicates that in higher dimensions segmentation and identification can be easier, possibly because a weaker overlap of manifolds in higher dimensions.

4 Prediction

The assumption of stationarity is problematic in many cases of data analysis. Our approach provides a diagnostic tool as well as a good predictive solution for problems

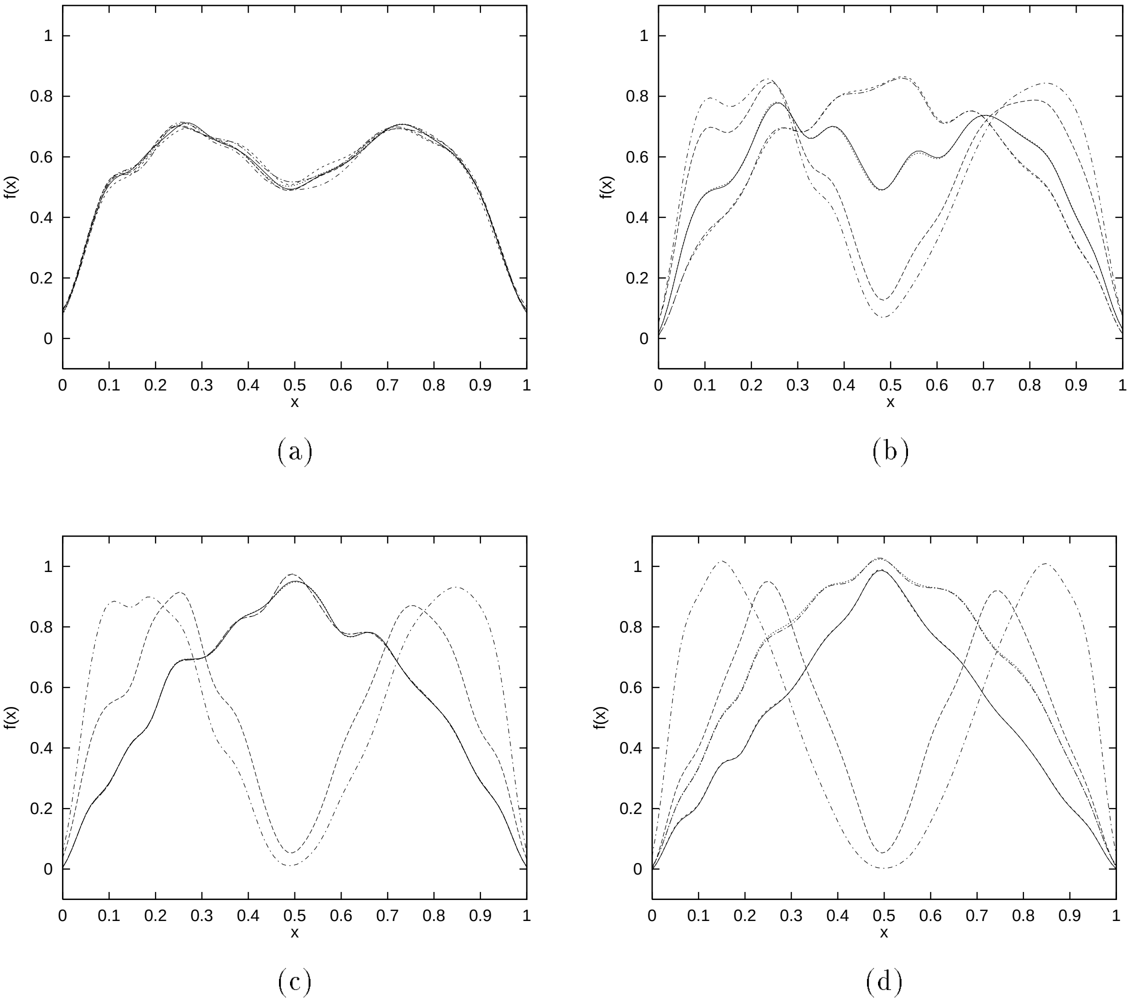


Figure 4: *Shown are the maps that have been learned by the predictors, (a) before the first and (b)-(d) after each of three phase transitions. The final result, after training has reached hard competition, is shown in Fig.3(b).*

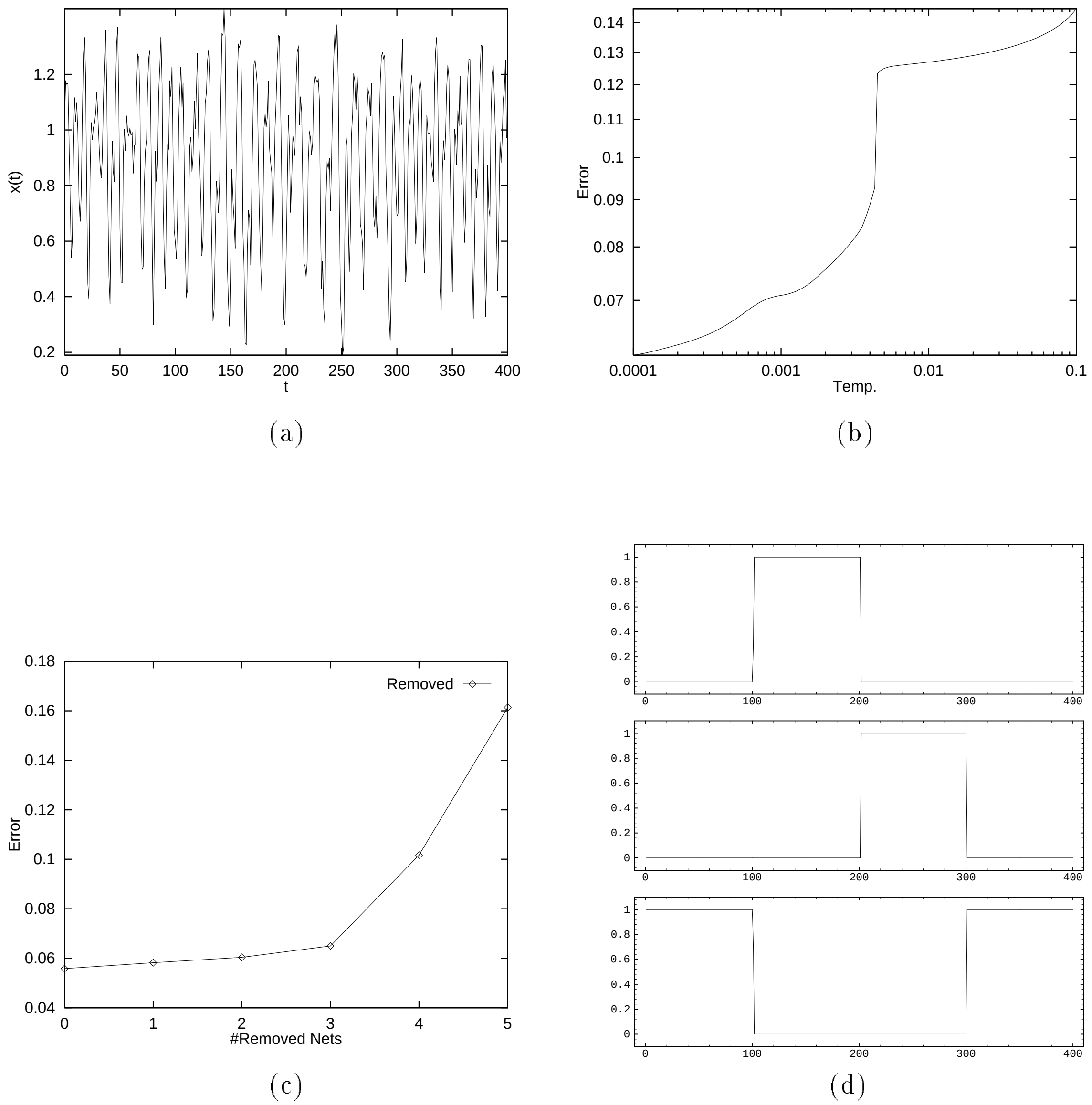
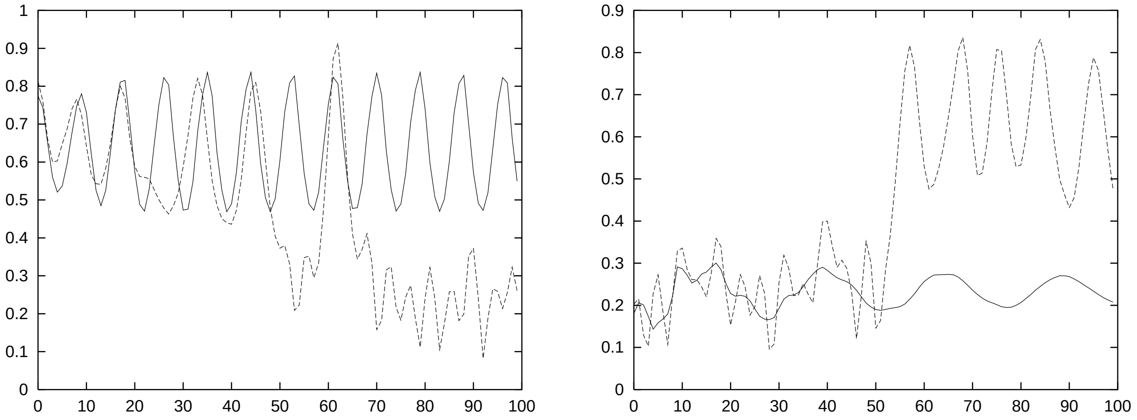


Figure 5: *(a) A noisy Mackey-Glass time series that includes 3 different dynamics was used for the segmentation task. (b) Adiabatic evolution of the training error for the Mackey-Glass data. (c) Increase of the prediction error when successively removing predictors (after training). Although no further training has been performed, up to three predictors can be removed without a significant increase of error. The three remaining predictors specialized on each of the dynamics present in the data, as indicated by the p_i^t's for each net, shown in (d).*

where non–stationarities are present due to random jumps of system parameters. In this section we demonstrate the relevance of our approach for the prediction of time series. Yet, we would like to stress, that although we obtain a very good prediction *within* two switchings, we do not solve the problem of predicting the next point in time where the system will most probably switch its state. For this, the statistics of switching has to be included into the model, but this obviously would make a much higher amount of data necessary.

We applied our method to the prediction of Data Set D from the Santa Fe Time Series Competition (Weigend and Gershenfeld 1994). This scalar data set was generated from a nine-dimensional periodically driven dissipative dynamical system with an asymmetrical four-well potential and a drift on the parameters. We used 6 RBF predictors that should predict a data point using 20 preceding points, i.e. the embedding dimension was $m = 20$. The training set was restricted to the last 2000 points of Data Set D to keep the computation time tolerable. After training was finished, the prediction of the training data was shared among the predictors.

The prediction of the continuation of Data Set D was simply done by iterating the particular predictor that was responsible for the generation of the latest training data. This predicted continuation was then compared to the true one – the test set –, which was originally unknown to the participants of the competition. Our method was quite useful for up to 50 time steps (see Fig.6(a)). After 50 steps, the system presumably performs a switch to another part of its potential, which per construction can not be foreseen by our approach, since the switching statistics has not been taken into account. Nevertheless, we tested the ability of this method to predict other parts of the test set by the other predictors and also found good performance up to about 50 time steps (Fig.6(b)). Again, we found that the prediction fails, when the system apparently jumps into a different state. Although the underlying system in this case was almost stationary, these results demonstrate that divide and conquer is a useful strategy here, because of the high dimensionality of the system and the complex form of the potential. A quantitative comparison with the winners of the Santa Fe Competition, Zhang and Hutchinson (Weigend and Gershenfeld 1994, pp. 219-241), demonstrates the power of our method. These authors applied a stationary approach that uses 100 hours of training time on a Connection Machine CM-2 with 8192 processors, and achieved a prediction error of 0.0665 (RMSE, root mean squared error) which they computed only for the first 25 step predictions, because their prediction broke down after that. Even if we compare our prediction only for this short episode, we find a RMSE of 0.0596, that is 10% better, and training took just two and a half hours on a SUN 10/20GX.



(a) (b)

Figure 6: *Prediction (solid line) of the continuation of Data Set D (dashed line) using the competing predictors approach. The predictors decompose the dynamics of the time series into simpler prediction tasks, so that each predictor is able to predict certain segments of the data (as shown in (a) and (b)). The accuracy for the first 25 step predictions is 10% better than the result of Zhang and Hutchinson, the winners of the Santa Fe Competition in 1991.*

Another well–known example of non–stationary dynamics in the real world is speech. Recently we also applied our method to predicting the dynamics of plain A/D-converted speech data. We find that the experts trained only on a single sentence, already reliably segment this signal, so that the unsupervised segmentation according to the dynamics can be used for word recognition. However, we did not observe a clear relation of the segmentation to the phonemes and we suspect that this requires a more careful choice of the experts, e.g. according to models of the vocal tract (for details we refer to Müller et al. 1995).

5 Summary and Outlook

We presented a new approach for the analysis of time series. It applies to systems, where non–stationarities are caused by switching dynamics. The two salient ingredients of our method are *memory* derived from a low switching rate (used in the mixing coefficients p_i^t) and an *adiabatic* enforcement of the competition during learning. We illustrated the performance of our approach with time series from alternating chaotic systems. In particular, we demonstrated that our approach is able to resolve ambiguities, which are present in the general case of overlapping input-output relations, with only very few assumptions about the systems generating the data, thereby leading to an unsupervised segmentation. The approach does not estimate the switching process itself, but serves as an analysis tool for the dynamics between switching events. The method is very robust, since it does not require *any* statistics of switching events. Also note, that our ansatz can nevertheless be used to obtain a model for the switching dynamics, once a valid segmentation is found.

We should also point out here, that the assumption of a low switching rate is essential to get the desired segmentation, at least when overlapping input domains are considered. This is due to the fact, that for a given data stream a variety of switching dynamical systems are conceivable as its origin. In our framework, the choice of models is a priori limited by the number of predictors, and the predictors we use only allow for relatively simple and smooth mappings. Nevertheless, it is still possible to fit the data in various ways; at least the training process is likely to select a wrong model and to get stuck in local minima of the error function. Constraining the training process to find only those models with a relatively low switching rate solves this problem (of course, only in those cases, where the dynamics does indeed switch at low rates). We do this, by imposing a low-pass filter on the errors. With this additional constraint, it is likely to obtain the correct segmentation together with appropriate models for the underlying sources.

This became evident when we compared our approach with the mixtures of controllers architecture (Cacciatore and Nowlan 1994). In this extension to the mixtures of experts (Jacobs et al. 1991), a Markov assumption about the switching characteristics is made in advance. A gating network is to *learn* the switching probabilities together with the dynamics of the sources. We tested the mixtures of controllers on the data presented in this paper. We found that this architecture only incidentally came up with the correct segmentation. In most cases it failed to converge to the correct model and ended up in a heavily switching solution. In contrast to that, our method always yielded the correct result.

Another problem arises for the mixture of controllers approach, when overlapping input domains are considered. Then, input sequences appear, that do not allow for a unique determination of the source. For *totally* overlapping input domains, as in our example, this is always the case. Since the gating network is triggered only by the input data, it receives no information about the operation mode and hence cannot produce a reasonable segmentation. Taking into account, that time series of switching dynamics with more or less overlapping input domains are the really challenging tasks[2], this appears to be a considerable disadvantage.

Two applications demonstrate the power of our approach: prediction of time series and segmentation of speech-data (presented in Müller et al. 1995).

When our approach is used to predict complex dynamics, the prediction quality can be improved significantly due to the divide–and–conquer strategy inherent in the ensemble of experts. In particular, we can significantly improve the results of the Santa Fe Prediction Competition (Weigend and Gershenfeld 1994) on data set D, which shows that this time series can efficiently be described as a switching dynamics.

Our future work will be dedicated to the application of this method to forecasting problems and to the classification of continuously spoken words. Further interest is also to estimate the dynamics of switchings in order to predict not only the inter-switch dynamics but also the dynamical changes themselves.

Acknowledgement: K.P acknowledges support of the DFG (grant Pa 569/1-1) and K. -R. M. acknowledges financial support by the European Communities S & T fellowship under contract FTJ3-004.

References

[1] Bengio, Y., Frasconi, P. (1994). Credit assignment through time: Alternatives to backpropagation. NIPS 93, Morgan Kaufmann.

[2] Cacciatore, T.W., Nowlan, S.J. (1994). Mixtures of Controllers for Jump Linear and Non–linear Plants. NIPS 93, Morgan Kaufmann.

[3] Casdagli, M. (1989). Nonlinear Prediction of Chaotic Time Series, Physica D **35**, 335-356.

[4] Jacobs, R.A., Jordan, M.A., Nowlan, S.J., Hinton, G.E. (1991). Adaptive Mixtures of Local Experts, *Neural Computation* **3**, 79-87.

[5] Kaneko, K. (1989). Chaotic but Regular Posi-Nega Switch among Coded Attractors by Cluster-Size Variation, Phys. Rev. Lett. **63**, 219.

[2] In one dimensional time series where the modes of the dynamics produce data in distinct domains, segmentation can be done by merely looking at the data.

[6] Kohlmorgen, J., Müller, K.-R., Pawelzik, K. (1994). Competing Predictors Segment and Identify Switching Dynamics. *Proc. of the International Conference on Artificial Neural Networks*, ICANN 94, Springer London, pp. 1045 ff.

[7] Liebert, W., Pawelzik, K., Schuster, H.G. (1991). Optimal Embeddings of Chaotic Attractors from Topological Considerations, Europhys. Lett. **14**, 521.

[8] Müller, K.-R., Kohlmorgen, J., Pawelzik, K. (1994). Segmentation and Identification of Switching Dynamics with Competing Neural Networks. ICONIP 94: Proc. of the Int. Conf. on Neural Information Processing, Seoul.

[9] Müller, K.-R., Kohlmorgen, J., Pawelzik, K., Analysis of Switching Dynamics with Competing Neural Networks, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, to appear October 1995

[10] Mackey, M., Glass, L. (1977). Oscillation and Chaos in a Physiological Control System, Science **197**, 287.

[11] McLachlan, G.J., Basford, K.J. (1988), *Mixture models*, Marcel Dekker, NY and Basel.

[12] Moody, J., C. Darken (1989). Fast Learning in Networks of Locally-Tuned Processing Units. *Neural Computation* **1**, 281–294.

[13] Pawelzik, K. (1994). Detecting coherence in neuronal data. In: Domany,E., Van Hemmen, L., Schulten, K., (Eds.), *Physics of neural networks*, Springer.

[14] Rabiner, L.R. (1988). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. IEEE, Vol **77**, pp. 257–286.

[15] Rose, K., Gurewitz, E., Fox, G. (1990). Statistical Mechanics and Phase Transitions in Clustering. *Phys. Rev. Letters*, Vol. 65, 945–948.

[16] Shamma, J.S., Athans, M. (1992). Gain scheduling: potential hazards and possible remedies. IEEE Control Systems Magazine, 12(3), 101–107.

[17] Takens, F. (1981). Detecting Strange Attractors in Turbulence. In: Rand, D., Young, L.-S., (Eds.), *Dynamical Systems and Turbulence*, Springer Lecture Notes in Mathematics, **898**, 366.

[18] Tong, H., Lim, K.S. (1980). Threshold autoregression, limit cycles and cyclical data. J. R. Stat. Soc. B **42**, 245–268.

[19] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K. (1989). Phoneme Recognition Using Time-Delay Neural Networks, IEEE Int. Conf. on Acoustics, Speech and Signal Processing.

[20] A.S. Weigend and N.A. Gershenfeld (Eds.) (1994). *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley.

[21] Yuille, A.L., Stolorz, P., Utans, J. (1994). Statistical Physics, Mixtures of Distributions, and the EM Algorithm, Neur. Comp. **6**, 334–340.