Fast Change Point Detection in Switching Dynamics using a Hidden Markov Model of Prediction Experts

J. Kohlmorgen*, S. Lemm*, K.-R. Müller*, S. Liehr‡, K. Pawelzik‡

* GMD FIRST, Rudower Chaussee 5, D-12489 Berlin, Germany
e-mail: {jek, lemm, klaus}@first.gmd.de

‡ Institute for Theoretical Physics, Kufsteiner Str., D-28334 Bremen, Germany
email: {sliehr, pawelzik}@physik.uni-bremen.de

Abstract

We present a framework for modeling switching dynamics from a time series that allows for a fast on-line detection of dynamical mode changes. The method is based on a hidden Markov model (HMM) of prediction experts. The predictors are trained by Expectation Maximization (EM) and by using an annealing schedule for the HMM state probabilities. This leads to a segmentation of the time series into different dynamical modes and a simultaneous specialization of the prediction experts on the segments. In a second step, an input-density estimator is generated for each expert. It can simply be computed from the data subset assigned to the respective expert. In conjunction with the HMM state probabilities, this allows for a very fast on-line detection of mode changes: change points are detected as soon as the incoming input data stream contains sufficient information to indicate a change in the dynamics.

1 Introduction

Modeling dynamical systems through a measured time series is commonly done by reconstructing the state space with time-delay coordinates [12]. The prediction of the time series can then be accomplished e.g. by training neural networks [13]. If, however, a system operates in multiple modes and the dynamics is switching between these modes, standard approaches like multi-layer perceptrons fail to represent the underlying input-output relations when attractor manifolds overlap. Moreover, they do not reveal the dynamical structure of the system.

In [6, 8, 10] we have described a framework (the ACE algorithm) for time series from switching dynamics, where an ensemble of neural network predictors specializes on the respective operating modes. We now extend this approach in two ways: (I) by using a more suitable hidden Markov assumption instead of assuming that switching does not occur in small time windows (the latter leads to a low-pass filter on the prediction errors [10]), and (II) by using input information *in addition* to error information from the output in order to decide which mode is present. We demonstrate that mode changes can be detected much earlier if the latest available input data is taken into account.

2 Prediction Experts in a Hidden Markov Model

In the following we assume that the reader is already familiar with the basic principles of hidden Markov models (HMMs). For a thorough introduction, we would like to refer to the tutorial of Rabiner [11], since we also make use of his notation.

Consider an HMM where each state is represented by a prediction expert, e.g. a neural network in case of non-linear dynamics. The HMM consists of (1) a set $S = \{s_i\}$ of states, (2) a matrix $A = \{a_{ij}\}$ of state transition probabilities, (3) an observation probability distribution $p(y|s_i)$ for each state s_i, which is a continuous density in our case, and (4) the initial state distribution $\pi = \{\pi_i\}$.

Each state $s_i \in S, i = 1, \ldots, N$, represents a prediction expert $f_i(\vec{x}_t)$ that predicts a future value $y_t = x_{t+\tau}$ of a time series $\{x_t\}$ given a vector of past values $\vec{x}_t = (x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau})$. The parameter d is called the embedding dimension and τ is called the delay parameter. Note that the extension to multivariate time series is straightforward.

Under a gaussian assumption, the probability that a particular predictor f_i would have produced the observed data y_t is given by

$$p(y_t \mid s_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y_t - f_i(\vec{x}_t))^2/2\sigma^2}. \quad (1)$$

This equation represents the observation probability distribution for each state, where we simply set σ^2 to the variance of the training data. Without any prior knowledge about the initial state distribution π, the states are assumed to be equally probable at the beginning of the time series, $\pi_i = 1/N$.

The transition matrix $A = \{a_{ij}\}$ determines the probability to switch from a state s_i to a state s_j. In principle, this matrix can be found using a training procedure, as e.g. the Baum-Welch method [11]. However, since we focus on problems with only relatively few switching events, the matrix is used to incorporate this prior knowledge in such a way that remaining in the current state is k times more likely than switching to another state (low switching rate assumption):

$$a_{ij} = \begin{cases} \frac{k}{k+(N-1)} & ; \text{if } i = j \\ \frac{1}{k+(N-1)} & ; \text{if } i \neq j \end{cases} \quad (2)$$

With the assumption about the switching rate in terms of a single parameter k ($k > 1$), we thus get a fixed transition matrix. We found that the restriction on models with a low switching rate is important [10], since it reduces the degrees of freedom in model space and effectively prevents from overfitting the data.

3 Training the Experts by Deterministic Annealing

The experts are trained by minimizing the Kullback-Leibler divergence

$$\mathrm{KL}(Q||P) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} Q_{i,t} \log \frac{Q_{i,t}}{P_{i,t}} \quad (3)$$

where $Q_{i,t} = \gamma_t(i)$ is the HMM probability of being in state i at time t given the hidden Markov model and all the training data, and $P_{i,t} = P(y_t, s_i) = P(y_t|s_i)P(s_i)$ is the probability of being in state i at time t just given the prior $P(s_i) = 1/N$ and a single training pattern $(\vec{x}_t, y_t)$.

The training can be performed efficiently by Expectation Maximization (EM) [1, 4]. The E-step consists in estimating the probabilities $Q_{i,t} = \gamma_t(i)$. These probabilities can be computed by the well-known *forward-backward* algorithm for hidden Markov models [11]. The M-step then adapts the model by minimizing the KL-divergence for the given $\gamma_t(i)$ using gradient descent. Since in the M-step the $\gamma_t(i)$ are considered to be constant, the derivative of the KL-divergence with respect to the output of an expert f_i (the learning rule) can

be simplified drastically:

$$\frac{\partial \mathrm{KL}}{\partial f_i} = \frac{\partial}{\partial f_i} \frac{1}{T} \sum_{t=1}^{T} Q_{i,t} \log \frac{Q_{i,t}}{P_{i,t}}$$

$$= \frac{\partial}{\partial f_i} \frac{1}{T} \sum_{t=1}^{T} Q_{i,t} \log Q_{i,t} - Q_{i,t} \log P_{i,t}$$

$$= \frac{\partial}{\partial f_i} \frac{1}{T} \sum_{t=1}^{T} -Q_{i,t} \log P_{i,t}$$

$$= \frac{1}{\sigma^2} \frac{1}{T} \sum_{t=1}^{T} -Q_{i,t} \; (y_t - f_i(\vec{x}_t)).$$

In order to achieve a hard segmentation of the time series and exclusively assign the data points to the experts, we introduce a deterministic annealing into the learning rule by using a soft-max function over $\gamma_t(i)$,

$$Q_{i,t} = \frac{e^{\gamma_t(i)/\theta}}{\sum_{j=1}^{N} e^{\gamma_t(j)/\theta}}, \qquad (4)$$

instead of using $\gamma_t(i)$ directly. In eq. (4), we adiabatically anneal the "temperature" parameter θ to zero during training, which finally leads to an exclusive assignment of training data points to experts. Moreover, the annealing of $Q_{i,t}$ also promotes the initial diversification of the experts. In the limit, $\theta \to 0$, at the end of the training phase, the annealed probabilities $Q_{i,t}$ yield the hard segmentation of the time series into dynamical modes, and the respective prediction experts f_i represent the individual dynamical systems.

4 On-line Detection of Change Points

Given the trained HMM, we can now perform an on-line segmentation and detection of change points on new, incoming data. Let us consider the case where we have already collected some data $\{x_t\}, t = 1, \ldots, T$, which e.g. might be the initial training data set. With each new incoming data point, $x_{T+1}, x_{T+2}, \ldots$, we can then compute a new segmentation: the new matrix of state probabilities $\gamma_t(i)$ and the corresponding hard segmentation of it, $Q_{i,t}$, with $\theta \to 0$. It turns out, however, that mode changes are not detected instantaneously. For two reasons: first, data from a new dynamical mode lies typically within the same region as data from the previous mode (otherwise segmentation would be pretty simple). It might therefore also belong to the previous mode with some probability. Second, the prior of a low switching rate lets the model prefer to remain in its current state rather than to switch to a new one.

In many applications, e.g. in finance or in safety-critical systems, it is however important to detect dynamic changes as soon as possible. The idea to improve the detection of change points is the following: Although the prediction error for the latest available data point, x_{t^*}, might not yet be sufficient to unambiguously identify the appropriate prediction expert after a mode change, the latest available embedding vector $\vec{x}_{t^*}$ might already be sufficient, since it represents a point in phase space and not just a projection of it. Therefore, instead of simply using the (a priori) criterion[1] $P(i_{t^*}) = \gamma_{t^*-\tau}(i)$ to decide which mode is present at time t^*, we propose to use

$$P(i_{t^*} \mid \vec{x}_{t^*}) = \frac{P(\vec{x}_{t^*} \mid i_{t^*}) P(i_{t^*})}{\sum_{j=1}^{N} P(\vec{x}_{t^*} \mid j_{t^*}) P(j_{t^*})},$$

which takes the latest available information into account. In the above equation, an input-density estimator $P(\vec{x}|i)$ is required for each expert i. To this end we use standard kernel smoothers,

$$P(\vec{x} \mid i) =$$

$$\frac{1}{M_i} \sum_{m=1}^{M_i} \frac{1}{(2\pi h_i^2)^{d/2}} \exp\left(-\frac{(\vec{x} - \vec{x}_i^m)^2}{2h_i^2} \right),$$

where $\{\vec{x}_i^m\}_{m=1}^{M_i}$ is the subset of the training data that was assigned to expert i at the end of the HMM training phase. For the kernel

[1] Note that $\gamma_t(i)$ for $t > t^* - \tau$ can not be computed here, since the required observations $y_t = x_{t+\tau}$, that is $x_{t^*+1}, x_{t^*+2}, \ldots$, are not yet available.

width we use $h_i = \sigma_i M_i^{-1/5}$, as suggested in [15, p. 306], where σ_i^2 is the sample variance.

As shown in the following section, the quantity $P(i_{t^*}|\vec{x}_{t^*})$ can in fact detect a change point significantly earlier than $P(i_{t^*})$.

5 Experimental Results

As an example, consider a high-dimensional chaotic system generated by the Mackey-Glass delay differential equation [7]

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-t_d)}{1+x(t-t_d)^{10}}. \quad (5)$$

It was originally introduced as a model of blood cell regulation. Three stationary operating modes, A, B and C, are established by using different delays, $t_d = 17, 23$ and 30, respectively. After operating 100 time steps in one of the three modes (with respect to a subsampling step size $l = 6$), the dynamics is randomly switched to one of the other modes (Fig. 1).

The HMM-based EM algorithm is applied to the first 3000 data points of the generated time series, using an ensemble of three radial basis function (RBF) predictors $f_i(\vec{x}_t)$, $i = 1, ..., 3$. The input to each predictor is a vector $\vec{x}_t$ of time-delay coordinates of the scalar time series $\{x_t\}$. The embedding dimension is $d = 6$ and the delay parameter is $\tau = 1$ on the subsampled data. The RBF predictors consist of 10 basis functions each, with adaptive widths and centers, as proposed in [9]. After training, net 1 has specialized on mode B, net 2 on mode A and net 3 on mode C, which can be seen in the final segmentation in Fig. 1. Next, we computed the input-density estimators in order to compare the different methods of change point detection.

We then used the trained ensemble for on-line segmentation of new data. Fig. 2 shows the result for $P(i_{t^*})$, $P(i_{t^*}|\vec{x}_{t^*})$, and the ACE algorithm [10]. As already mentioned, the latter approach employs a low-pass filter on previous prediction errors as criterion to determine the current mode. The mode change in Fig. 2 actually takes place between $t = 100$ and 101 of the new, incoming data, where the system switches from mode A to mode B. Therefore, for the first 5 time steps (i.e. $d\tau - 1$) after the switching ($t = 101, \ldots, 105$), the embedding vectors $\vec{x}_t$ neither lie on the attractor of mode A nor on the attractor of mode B, because they include data points from both modes. Only after $t = 105$ there is consistent data that allows to identify mode B.

Using $P(i_{t^*})$, the switch is detected at $t = 117$: after $t = 117$ the predictor for mode B (solid line) is more likely for producing the data, whereas before $t = 117$ the predictor for mode A (dashed line) is more likely. The plot for $P(i_{t^*}|\vec{x}_{t^*})$ shows that the mode change is found earlier, the switching to mode B is indicated at $t = 107$. Thus, the mode change is detected very fast, considering the fact that mode A is actually left at $t = 100$ and mode B can only be identified after $t = 105$. Note that the low-pass filter of the ACE method yields a similar result as $P(i_{t^*})$, it detects the mode change likewise at $t = 117$. Like $P(i_{t^*})$, the ACE probabilities do not take the latest available input vector $\vec{x}_{t^*}$ into account.

6 Discussion

We presented a new method for the unsupervised segmentation and identification of switching dynamics. In contrast to previous algorithms [6, 8, 10], which are primarily designed for off-line analysis, it is based on a hidden Markov model and allows for a fast detection of change points in on-line scenarios. This is particularly important in financial or safety-critical applications. Moreover, in contrast to the Mixtures of Experts approach [5] or related ensemble methods [2, 3, 14], the presented framework (1) allows to incorporate prior knowledge about the switching probabilities in a straightforward way, (2) ensures a clear-cut subdivision of the training data into individual training sets for the experts, and (3) subdivides the training process into two stages, where the

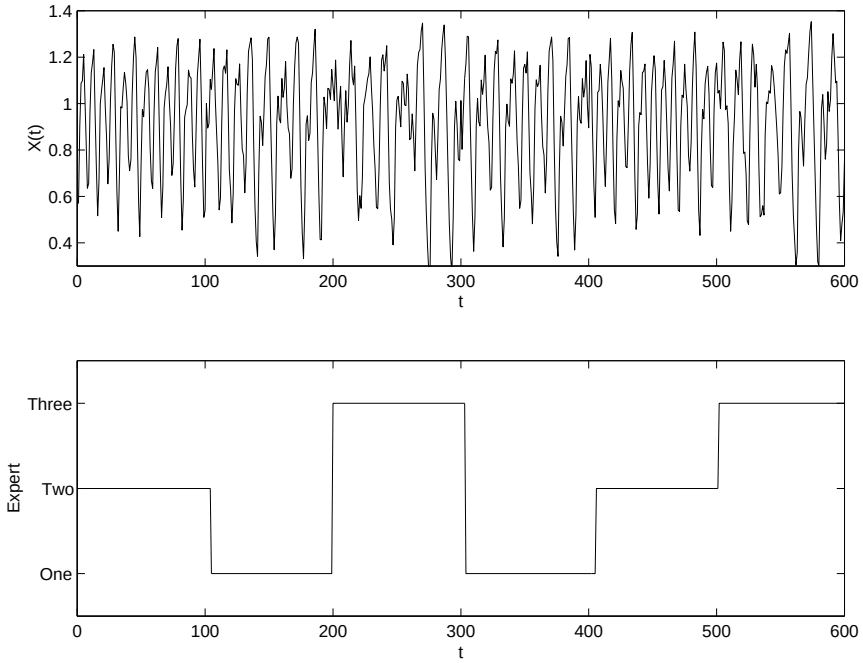

Figure 1: Top: The first 600 data points of the switching Mackey-Glass system (training data). Bottom: The obtained hard segmentation of the data after training. All dynamical modes are correctly identified and represented by three prediction experts.

second stage is not even an adaptation process but simply a *computation* of density estimators. These features substantially simplify the overall training task.

Our future work is dedicated to on-line change point detection in financial data. We expect, however, that this method will also be useful in other application domains.

Acknowledgement

We acknowledge support of the Deutsche Forschungsgemeinschaft (grants Ja379/51 and Pa569/2-1).

References

[1] Baum, L., Petrie, T., Soules, G., Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, **41**:164–171.

[2] Bengio, Y., Frasconi, P. (1995). An Input Output HMM Architecture. In *NIPS '94: Advances in Neural Information Processing Systems* 7, Morgan Kaufmann.

[3] Cacciatore, T.W., Nowlan, S.J. (1994). Mixtures of Controllers for Jump Linear and Non-linear Plants. In *NIPS '93: Advances in Neural Information Processing Systems* 6, Morgan Kaufmann, 719–726.

[4] Dempster, A., Laird, N., Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. Series B*, **39**:1-38.

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

[6] Kohlmorgen, J., Müller, K.-R., Pawelzik, K. (1995). Improving short-term prediction with competing experts. ICANN'95, EC2 & Cie, Paris, 2:215–220.

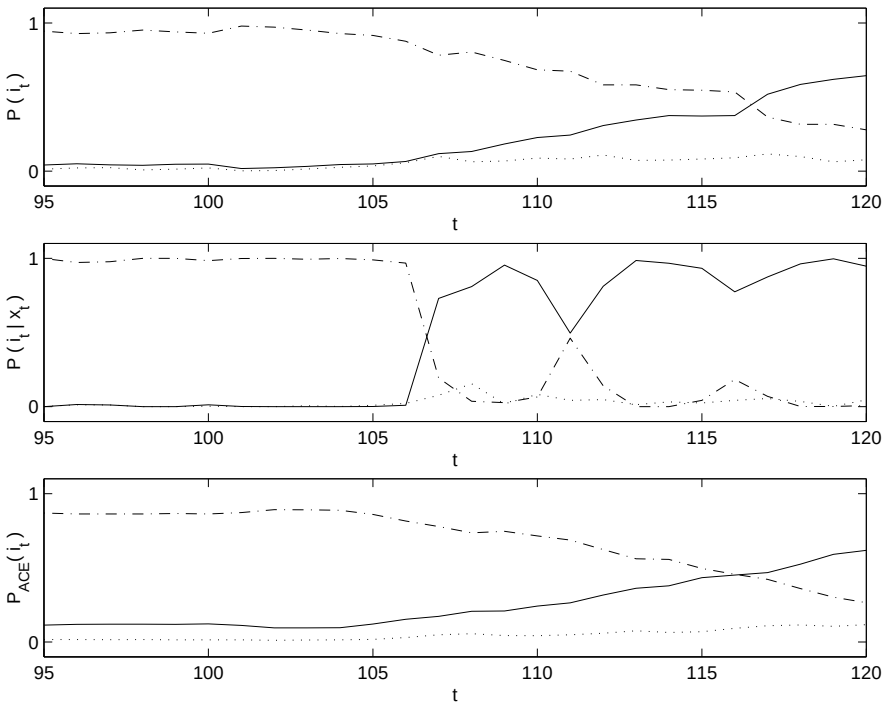

Figure 2: The on-line computed state probabilities for three states, using three different criteria: $P(i_{t^*})$, $P(i_{t^*}|\vec{x}_{t^*})$, and ACE. Clearly, $P(i_{t^*}|\vec{x}_{t^*})$ detects the mode change, which is actually at $t = 100$, much faster than the two other methods, which both yield similar results. Note that the new operating mode can only be identified after $t = 105$ (see text).

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

[8] Müller, K.-R., Kohlmorgen, J., Pawelzik, K. (1995). Analysis of Switching Dynamics with Competing Neural Networks, *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science*, E78-A, No.10, 1306–1315.

[9] Müller, K.-R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V. (1998). Using Support Vector Machines for Time Series Prediction, In: *Advances in Kernel Methods — Support Vector Learning*, eds. B. Schölkopf, C. Burges, A. Smola, MIT Press, Cambridge, MA.

[10] Pawelzik, K., Kohlmorgen, J., Müller, K.-R. (1996). Annealed Competition of Experts for a Segmentation and Classification of Switching Dynamics, *Neural Computation*, **8:2**, 342–358.

[11] Rabiner, L.R. (1988). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Readings in Speech Recognition*, ed. A. Waibel, K. Lee, 267–296. San Mateo: Morgan Kaufmann, 1990.

[12] Takens, F. (1981). Detecting Strange Attractors in Turbulence. In: Rand, D., Young, L.-S., (Eds.), *Dyn. Systems and Turbulence*, Springer Lect. Notes in Math., **898**, 366.

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

[14] Weigend, A.S., Mangeas, M. (1995). Nonlinear gated experts for time series: discovering regimes and avoiding overfitting, *International Journal of Neural Systems* **6**, 373–399.

[15] Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*, Springer, NY.