Bayesian techniques for modelling dynamic patterns

Alexander Ypma[1], Machiel Westerdijk[2], Henk-Jaap de Walle[3] and Tom Heskes[1]
[1] SNN, Geert Grooteplein 21, 6525 EZ, University of Nijmegen
[2] Cap Gemini Ernst & Young, P.O. box 2575, 3500 GN Utrecht
[3] BrandmarC, P.O box 135, 3820 AC Leusden
The Netherlands
E-mail: {ypma,tom}@snn.kun.nl
Web: www.snn.kun.nl/nijmegen/graphmod.html

Abstract

We give a short description of the Bayesian approach to adaptive data modelling. Then we demonstrate the modelling process in two real-world applications with dynamic data.

1 Introduction

An important issue in data modelling is how prior knowledge can be combined with data. One can already 'steer' the solution based on domain knowledge, and tune the specifics of the model with the observed data. The risk of overtraining to a particular dataset can be diminished and more meaningful models may be obtained. Typical data modelling tasks are clustering, classification, regression, projection and density estimation. A good model for a system captures prior expectations on the data and makes good predictions of new data from the system. We give a short description of the Bayesian approach to adaptive data modelling. Then we demonstrate the modelling process in two real-world applications with dynamic data.

2 Framework for Bayesian modelling

Bayesian modelling relies on Bayes' rule of statistical inference:

$$\begin{aligned} P(w|D) &= \frac{P(D|w)P(w)}{P(D)} \\ \text{posterior} &= \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \end{aligned}$$

where the normalizer equals $P(D) = \int P(D|w)P(w)dw$ Application of this rule can be looked upon as a general mechanism to combine prior knowledge $P(w)$ on the model parameters w with the data likelihood $P(D|w)$ into a posterior distribution over the parameters after the data has been observed. Unfortunately, the normalising constant is often an intractable quantity. In these cases, approximate posteriors may be formulated that are tractable and informative. Note that full Bayesian inference leads to confidence levels on the parameters, rather than a point estimate. The Bayesian modelling approach comprises the following stages [Mac03]: model fitting, model comparison, and prediction.
1. Model fitting: we define a set of model structures $\mathcal{H} = \{H_j\}, j = 1, \ldots, M$. Now, assume H_i is true, and learn model parameters w given data D

$$P(w|D, H_i) = \frac{P(D|w, H_i)P(w|H_i)}{P(D|H_i)}$$

If full Bayesian inference of the posterior is troublesome or too time demanding one can search for the *most probable* a posteriori (MAP) parameters:

$$w_{\mathrm{MAP}} \quad = \quad \underset{w}{\operatorname{argmax}}\, P(w|D, H_i)$$

Note that the intractable normaliser does not have to be computed any more. The maximum likelihood (ML) estimate is obtained if the prior is not taken into account.
2. Model comparison: infer which model $H_i \in \mathcal{H}$ is most plausible given D

$$P(H_i|D) \quad \propto \quad P(D|H_i)P(H_i)$$

Here, the evidence for the model is

$$P(D|H_i) \quad = \quad \int P(D|w, H_i)P(w|H_i)dw$$

which does not depend on the model parameters (they are integrated out) but is a function of the *model structure* and the data only. It can be used to compare the suitability of different model structures for the data, e.g. should we use 4 or 5 hidden units in a neural network model?
3. Prediction: weigh the predictions of each model with the likelihood of the model; sum all weighted predictions. Proper Bayesian prediction uses all models ('hypotheses about the data') for the prediction and emphasizes models with higher model evidence. A proxy to this way of predicting is to choose the structure with highest evidence and use its MAP parameters in the prediction. This still bears some risk of overfitting, though this risk is diminished by using the evidence (that will penalise unsuitable model structures) and a prior.

2.1 Bayesian belief networks

The former strategy can be exploited for modelling the dependencies between variables in a certain domain. A *Bayesian belief network* or *probabilistic graphical model* is a graph depicting the probabilistic relations between variables. Each arrow implies a parametric stochastic dependency; nodes may be unobserved (they represent a conditional probability distribution over the variable) or observed (where they are clamped to a certain value). As an example, in figure 1 the well-known *sprinkler* example from [Pea97] is drawn as a graphical model. The probability that the grass is wet depends both on the probability of rain and on the probability that the neighbour's sprinkler is on. These two variables in turn depend on the probability of cloudiness. Once we observe that the grass is wet, our prior beliefs about the 'sprinkler' and the 'rain' variable are updated with the evidence on the 'wet grass' node. If we then observe that e.g. it is very cloudy, we can infer by Bayes' rule that the posterior on 'sprinkler' will be decreased while the posterior on 'rain' will be increased. Graphical models exploit the conditional independencies between variables: if an arrow is absent between two nodes they are conditionally independent. In many domains it will be possible to represent the problem in terms of only locally dependent variables, so this will ease the specification of prior knowledge. Furthermore, probability distributions over many variables become quickly intractable; being able to split these distributions in conditional distributions over less variables will make the computations more tractable. Also, when there are only weak dependencies between subsets of variables, we can *approximate* the full distribution with a set of uncoupled subsets, and make inference even more tractable without loosing much accuracy.

2.2 Learning a graphical model

In this paper we will use the typical procedure for MAP or ML learning of graphical models, called Expectation-Maximization or E-M, [DLR77]. This is not a full Bayesian approach, so regularisation (avoiding overfitting by using prior knowledge on the structure or the parameters) is very important when making models from data.

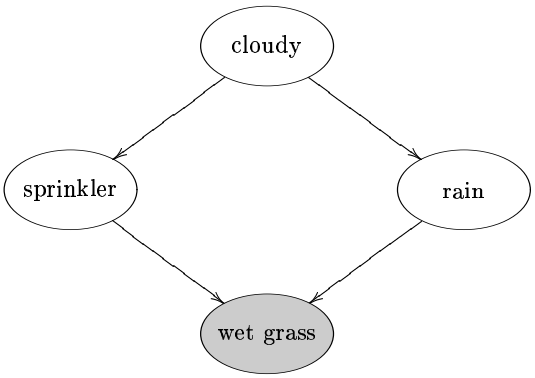

Figure 1: Graphical model of the *sprinkler* problem from [Pea97].

1. specify (or learn) suitable structure
2. learn MAP (or ML) parameters:
   - (E)xpectation: infer values of hidden variables, and
   - (M)aximization: find MAP (or ML) parameters using this estimate.

The first stage deals with the model structure. We should either specify it using domain knowledge, or apply (greedy) algorithms for learning which dependencies should be present in our model ('structure learning'). In the second stage, we learn the model parameters by repeatedly estimating the posterior distributions over de hiddens, given the current parameters (E-step) and estimating the 'best' parameters, given the current estimate of the posterior over the hiddens (M-step). This algorithm is guaranteed to end up in a (local) maximum of the likelihood (ML) or parameter posterior (MAP), figure 2.

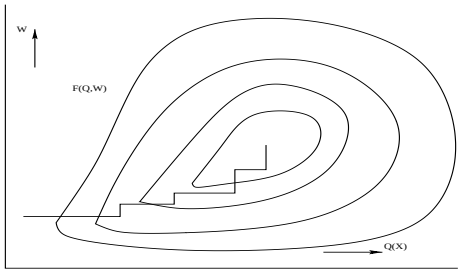

Figure 2: Execution of the E and M steps can be looked upon as taking orthogonal steps in the space of parameters and hidden posteriors, approaching the (local) optimum of the cost function more closely during each iteration.

3 Heterogeneous symbolic dynamic data

In many applications with transaction data one will encounter symbolic sequences of unequal length. A symbol may be e.g. a bank account, a telephone number, a web page or a hospital transaction. A raw datafile can then be mapped onto a collection of symbolic sequences. For example, three different sequences of two different users might be represented as

```
seq1 = 8 9 10 11 11 11 seq2 = 1 2 4 3 5 7 6 5 7 6 4 3

seq1 = 8 8 9 12 13 13 14 15 14 8 9 seq2 = 1 2 4 3
```

seq1 = 8 9 21 22 seq2 = 1 2 2 2 2 4 2 4 3

An interesting question is now how to assign a newly observed path

seq? = 8 9 9 10 11 11 153 154 155 9 9 9 8 9 10 11 11 11 11 11 24

In order to be able to make quantitative statements about the 'fit' of the last sequence to the previous ones, we have to use a probabilistic generative model for the data. In this model we make the following assumptions. Each user i behaves like one of C user-groups; each group shares common (static) characteristics S. A user i produces n_i transaction streams of length $T_{ij}, i = 1, \ldots, N, j = 1, \ldots, n_i$.

3.1 Model

We proposed the following graphical model for heterogeneous transaction data, see figure 3. It is a so-called *mixture of hidden Markov models*, with possibly additional *static* information (demographic, financial, etc.) on the user. In this model, $C \in \{1, \ldots, K\}$ is the (unobserved) cluster label, Π^k is the

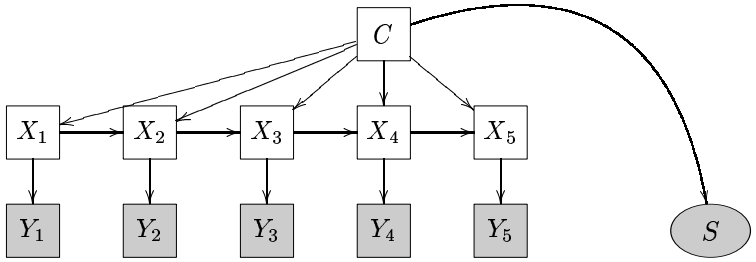

Figure 3: Mixture of hidden Markov models and static data, for heterogeneous symbolic time series.

prior for cluster k, A^k is the state-to-state transition matrix for k, B^k is the observation matrix for k, X_t is the (unobserved) state at time t, Y_t is a dynamic observation at time t and S is a static observation vector. Two examples of how the model can be applied are:

- Web mining [YH03]: X_t denotes page categories, Y_t denotes pages
- Hospital data mining: X_t denotes code categories ("care profile classes" (CPC) like 'diagnostic', 'lab', 'nursing', 'surgery', etc.), Y_t denotes transaction codes. In the case study below we actually observe the care profile classes with each transaction, so the model reduces to a mixture of Markov chains (mMC).

3.2 Using prior knowledge

Prior knowledge on the dynamics can be taken into account in the following manner [RSC02]. Consider a reestimated transition probability of the form $\hat{P}(i,j) = n_{ij}/n_i$, with n_{ij} the transition count from state i to j and n_i the number of transitions from i. If our prior knowledge takes the form of an additional pseudo-sequence of length $\beta + 1$, which is divided into β_{ij} transitions from i to j, the Bayesian MAP estimate is

$$\hat{P}_{\mathrm{MAP}}(i,j) = \frac{n_{ij} + \kappa\beta_{ij}}{n_i + \kappa\beta_i}, \tag{1}$$

where $\beta_i = \sum_j \beta_{ij}, n_i = \sum_j n_{ij}$ and $0 \leq \kappa \leq 1$ determines the extent to which the prior or the data is used. A similar trick can be applied to the 'prior probability' Π over states and the observation probabilities. Especially the latter quantity may easily tend to zero in cases with small sample sizes (limited number of observations, large dimensionality of the observables). From figure 3 it is clear that each mixture component has a private observation matrix. However, in for example a web mining

application the 'interpretation' of a category should preferably not be too different for different user types (e.g. 'Mercedes-Benz.html' should be categorized as 'cars', regardless the user's interests). Therefore, one can constrain the observation matrices in all clusters to be equal (a.k.a. *parameter tying*). This has the additional advantage that one decreases the danger of overfitting in cases with a large number of observables (i.e. sites with many pages and relatively small number of visitors).

3.3 Application to hospital transaction data

The financing system of health care in the Netherlands is currently transformed dramatically with the purpose to create a product driven open market. Here, hospitals and insurers will negotiate about price and delivery volume of health care products. The full set of health care products has been defined under supervision of Cap Gemini. The set consists of thousands of so called diagnosis-treatment combinations (DBCs), e.g. '11.1801.41: arthrosis of hip - surgery with clinical episode. In order to make the negotiation process feasible and to stimulate competition between hospitals this set had to be clustered into a smaller set of product groups, about 30 per medical specialism. The DBCs in one cluster have in common that the underlying care process is similar for each code, i.e. they are similar in the amount of resources used in each hospital department such as the lab, the operating room, intake, nursery, etc. The basic set of care processes were successfully identified by applying a clustering algorithm on data from 40 hospitals containing 1.5 million patient records [WLPM03]. For the purpose of financing we did not take dynamic effects into account. In an open market hospitals will be forced to apply more sophisticated measurement and control systems in order to work more cost efficient than the competitors. For this purpose Cap Gemini is currently developing instruments to optimize, amongst others, planning and scheduling of hospital activities. Applying the same philosophy as above, we expect that for planning and scheduling it is much more efficient to focus on the basic set of care processes, a small set, than on the individual diagnosis of which there are thousands. Of course, for planning and scheduling time and temporal ordering is crucial. As a first step we analyzed data from 4000 urology patients. Hospital activities were summarized in a few activity classes: lab (lab), ambulant (poli), surgery (oper), diagnostic (diag), nursery (klin), day care (dagb), therapeutic (ovtv). The amount of resources used in a class for one patient is indicated with an increasing number, e.g. lab0, lab1, lab2. Hence the care process associated with each patient is a series of activities, like (poli 1 $\rightarrow$ lab 2 $\rightarrow$ diag 3 $\rightarrow$ oper 1 $\rightarrow$ lab 2) which are ordered according to the day the activities took place. We do not yet have access to more accurate time scales. On this data set we first performed some preprocessing. All same CPC events on one day were lumped into one event, with aggregate cost; afterwards, the cost level was split into 2 or 3 levels (e.g. low, medium, high cost), giving a new 'aggregate CPC sub-symbol'. We selected the first 4000 patients (60 % train, 40 % test), and trained mixtures of Markov chains (in which each chain can be represented with a transition matrix and a vector of prior state probabilities). Using cross validation with 10 repetitions we found that the data can be represented with about 7 clusters without overfitting the data, see figure 4, left subfigure. The transition matrices of two of these clusters are visualised in figure 5 the prior state

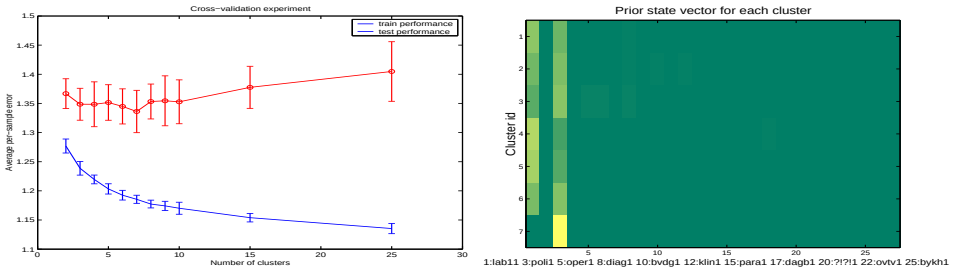

Figure 4: Left: crossvalidation experiment, using 10 repetitions. The top graph is the estimated generalisation error. Right: prior state vectors for the 7-state mixture of Markov chains.

vector can be seen in figure 4, right subfigure. There are 27 states, for example 1 = lab - low cost, 2 =

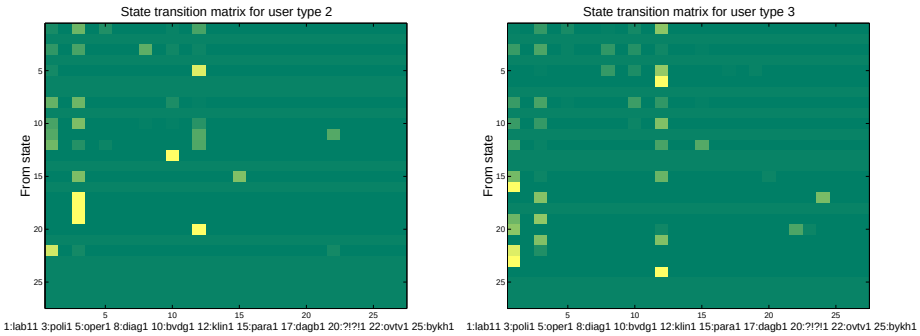

Figure 5: Transition matrices for two of the seven mixture components, i.e. clusters 2 (left) and 3 (right).

lab - high cost, 3 = poli - low cost, 6 = oper - medium cost, etc. From figure 4, right subfigure we see that the care process starts in either lab 1 or poli 1. In general we see (figure 5) that there is a strong tendency to jump to poli, lab and klin (cluster 2 and 3) followed by diag and bvdg.

4 Nonlinear dynamic processes

Many real-world systems are nonlinear, dynamic and stochastic in nature. Inference and learning of nonlinear system models with hidden dynamics is a difficult task, which requires approximations and simplifications to be made.

4.1 Model

Here we consider dynamical systems with nonlinearities in the state- and observation equations,

$$\begin{aligned} x_t &= f(x_{t-1}) + v_t, \quad v_t \sim \mathcal{N}(v_t; 0, \Gamma) \\ y_t &= g(x_t) + w_t, \quad w_t \sim \mathcal{N}(w_t; 0, \Sigma) \end{aligned} \tag{2}$$

where $f(\cdot)$ and $g(\cdot)$ are nonlinear functions, see figure 6, and $\mathcal{N}(x; \mu, \Sigma)$ denotes the normal distribution over x with mean μ and covariance matrix Σ. The graphical model for this system is shown in figure 6. The tasks in MAP or ML learning of the model are: infer the hidden states x_t; learn the parameters

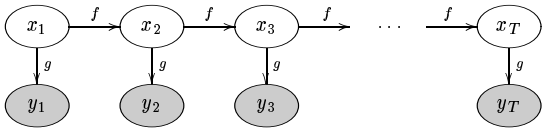

Figure 6: Nonlinear dynamical system. All nodes are continuous-valued, and f and g are arbitrary nonlinear functions. Shaded nodes are observed. Time progresses from left to right.

of $f, g,$ noise. In previous work [YH04] we developed algorithms to this end, and they are applied in the case study below. For the inference task, we make use of the unscented transform, which is a method to compute moments from nonlinearly transformed variables. Parameter learning can be done with E-M or with a related algorithm called Expectation-Conjugate Gradient. Here we parameterised the nonlinear functions with radial basis functions, similar to [RG01].

4.2 Application to marketing data

A marketer wants to know what drives sales, market share etc of a brand. In the literature you can find a lot of exercises modeling relations between price, promotions and market share. However there is

little to find about the implementation of a theoretical model as described in Brands and Advertising: How advertising effectiveness influences brand equity [FGH+99]. This model (see figure 7) describes not only a direct relation between marketing mix tools (price, promotions advertising etc) and sales, market share (this weeks promotion will increase sales by x %). Marketing mix instruments can also influence Top of Mind Awareness and perceptions like quality, price values. The awareness and perceptions itself are drivers for sales, market share next weeks (so if TOMA improves this week, this might increase sales in following weeks). Although theoretically known, this model lacks a quantitative estimation of

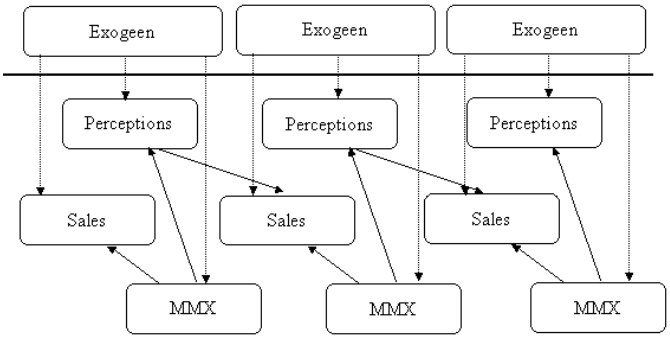

Figure 7: A model of market dynamics

its relations. In this paper we use two datasets. The first dataset consist of 5 input marketing mix investments and 5 exogenous variables. The output is 20-38 perception indicators and the sales figures of 5 shops. The measurement is on weekly basis. The second dataset contains of 8 marketing mix investments, 4 exogenous variables. In the second dataset the output is market share and 20 perception variables. Measurement is again on a weekly basis. The intuition on the hidden variables is that they represent the 'current opinion' or 'global trends'. We expect that a marketing steering variable has both an immediate influence on the output (via the observer) and a delayed influence via the dynamics (e.g. when 'the general opinion' about a brand gradually changes as a result of PR activities). Nonlinearities may enter the process because of sudden nonobserved disturbances and saturation effects. Two time series were 'compressed' into a 2-D hidden representation. The first time series contains of 12 inputs (marketing mix, exogenous), 21 outputs (market shares, consumer perceptions), length 64 weeks. The second time series contains 10 inputs (marketing mix, exogenous), 46 outputs (sales figures, consumer perceptions), length 24 weeks. The market-shares time series has periodicities in the order of 16 weeks

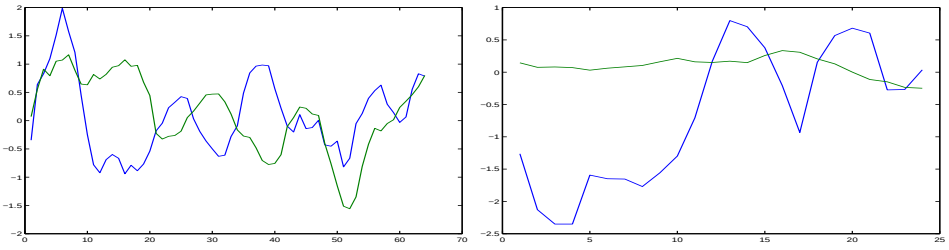

Figure 8: Compressed 2-D representation of marketing time series.

(figure 8, left subfigure), indicating more global trends. The sales time series shows underlying bursts (figure 8, right subfigure) that appear to be correlated with some of the inputs, indicating stronger dependence on steering variables.

5 Conclusion

We reviewed the Bayesian approach to adaptive data modelling. It provides a comprehensive and principled way to combine prior domain knowledge with data.
In the hospital data case study, we found that it is possible to find a compact representation of the set of hospital activity logs in terms of a small number of Markov chains. The probabilistic approach is crucial here regarding the enormous variability in care paths. However, the results also indicate that the date of the log does not accurately correspond to the date where the activity actually took place, e.g. we expect that care processes start with an ambulatory activity and not in the lab. To improve the model we first need more realistic time records. Given sufficiently accurate data, our approach will be to develop increasingly complex models, which represent higher order temporal dependencies and which represent the variability in time intervals between activities. Having found such a dynamic probabilistic representation, the idea is to start simulating future scenario's by changing parameters from their past value to see which setting leads to optimal performance.

In the marketing data case study we postulated hidden Markovian dynamics and learned an internal representation of two high-dimensional time series from marketing research. In the sequel we will study ways to incorporate prior knowledge on the process in the NLDS model and evaluate the predictive power of our method. One may postulate that there may be different underlying dynamical processes with different time constants and delay. E.g. an underlying 'trend' with relatively fast dynamics could be represented by one variable, whereas an 'image' variable could have slower time constants. This would lead to a model with separate types of dynamics, and differing level of nonlinearity in the transitions and observations.

6 Acknowledgements

This project was supported by the Dutch Technology Foundation STW, project NNN.5321 'graphical models for data mining'.

References

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1–38, 1977.

[FGH+99] Giep Franzen, Cindy Goessens, Mary Hoogerbrugge, Cees Kappert, Reint Jan Schuring, and Marnix Vogel. *Brands and Advertising: How advertising effectiveness influences brand equity.* 1999.

[Mac03] D. J. C. MacKay. *Information theory, inference, and learning algorithms.* Cambridge University Press, 2003.

[Pea97] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, 1997.

[RG01] S. Roweis and Z. Ghahramani. *An EM algorithm for identification of nonlinear dynamical systems*, chapter in Kalman Filtering and Neural Networks, Haykin (ed.). Wiley, 2001.

[RSC02] M. Ramoni, P. Sebastiani, and P. Cohen. Bayesian clustering by dynamics. *Machine learning*, pages 91 – 121, 2002.

[WLPM03] M.J.D. Westerdijk, M. Ludwig, S. Prins, and B. Misere. Building the dbc product structure. In *Patient Classification Systems Europe (PCSE)*, 2003.

[YH03] A. Ypma and T. Heskes. *O. R. Zaane, J. Srivastava, M. Spiliopoulou, B. Masand, WEBKDD 2002: Mining Web Data for Discovering Usage Patterns and Profiles*, volume 2703 of *Lecture notes in Artificial Intelligence*, chapter Automatic categorization of web pages and user clustering with mixtures of hidden Markov models (extended version), pages 35–49. Springer-Verlag, 2003.

[YH04] A. Ypma and T. M. Heskes. Novel approximations for inference and learning in nonlinear dynamical systems. In *Proceedings of 12th European Symposium on Artificial Neural Networks ESANN'2004*, 2004.