in: KI 2009 - Advances in Artificial Intelligence (KI), Paderborn, LNCS 5803, pp. 484–491, Springer Verlag 2009

Maximum a Posteriori Estimation of Dynamically Changing Distributions

Michael Volkhardt, Sören Kalesse, Steffen Müller, and Horst-Michael Gross*

Neuroinformatics and Cognitive Robotics Lab,
Ilmenau University of Technology, Germany
{michael.volkhardt,steffen.mueller,
horst-michael.gross}@tu-ilmenau.de
http://www.tu-ilmenau.de/neurob

Abstract. This paper presents a sequential state estimation method with arbitrary probabilistic models expressing the system's belief. Probabilistic models can be estimated by Maximum a posteriori estimators (MAP), which fail, if the state is dynamic or the model contains hidden variables. The last typically requires iterative methods like expectation maximization (EM). The proposed approximative technique extends message passing algorithms in factor graphs to realize online state estimation despite of hidden parameters. In addition no conjugate priors or hyperparameter transition models have to be specified. For evaluation, we show the relation to EM and discuss the transition model in detail.

Key words: state estimation, Bayesian filter, MAP, EM, factor graph, conjugate prior

1 Introduction

Probabilistic modeling techniques provide an appropriate tool set, when dealing with uncertainties in arbitrary systems. When tracking system states, probability theory models the system's belief and defines the required base operations for state estimation [1]. Recently, graphical models have been established as a powerful tool to visualize probabilistic models, whereat the influence of graph theory allows efficient algorithms for probabilistic inference [2], [3], [4], [5]. As described subsequently, this work extends factor graphs, which provide a powerful representation of graphical models for inference by explicitly modeling the model variables and their dependencies [6]. The exchange of messages in acyclic factor graphs is made possible by sequential message passing [7], while loopy belief propagation provides a message passing scheme for cyclic factor graphs [8].

* This work is partially supported by EU-FP7-ICT Grant #21647 to H.-M. Gross. M. Volkhardt, S. Müller, H.-M. Gross are with Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, 98684 Ilmenau, Germany michael.volkhardt@tu-ilmenau.de

Our aim is the tracking of uncertain system states modeled by arbitrary complex probability distributions (discrete, continuous or mixed). These can be expressed by graphical models or factor graphs, respectively. The parameters of stationary probability distributions can be learned from data samples using maximum likelihood estimation (ML), maximum a posteriori estimation (MAP) or expectation-maximization algorithm (EM). ML fits a parameter set of a probabilistic model to a given data set by solving an optimization problem. MAP augments ML with a conjugate prior distribution on the unknown parameters. By using Bayes' theorem, this distribution can be adapted sequentially with new observations [9]. If the model depends on unobserved latent variables, a sequential estimation is not possible and MAP estimation fails. EM overcomes this problem by iteratively calculating the hidden parameters in a first step and selecting the parameters of interest with ML in a second step in order to maximize the likelihood of the data [10]. Unfortunately, EM is not a sequential method and therefore requires the complete data set of observations.

MAP as well as EM fail, if the probability distribution changes permanently. When using Bayesian filtering for tracking states expressed by complex distributions, it is very hard to model the transition model and the conjugate prior of the system's state. This paper addresses these problems and implements an online message passing algorithm in extended factor graphs that allows approximative state tracking.

The remainder of this paper is organized as follows. Section 2 describes problems that occur while estimating a system's state represented by a complex probability distribution. To solve these problems, dynamic MAP estimation in extended factor graphs is presented in Sect. 3. For evaluation we show the relation of the developed method to EM algorithm. The paper concludes with a discussion on the assumptions and limitations of the algorithm and a summary.

2 Bayesian Filtering of complex probability distributions

We assume an exemplary system state modeled by a factor graph with random variable X that is dependent on an unobserved variable Y. The belief on the system's state hence is defined by factor potential $p(\mathbf{Z}) = p(X|Y)$. This conditional probability distribution is defined by parameters $\mathbf{\Theta}$. To estimate the unknown system's state, we search for probability distribution $p(\mathbf{\Theta})$.

MAP estimation can be applied to sequentially update the unknown probability distribution $p(\mathbf{\Theta})$ if distribution $p(\mathbf{Z})$ is constant and all variables are observed. In the context of state estimation, the problem of a permanently changing probability arises. By introducing a transition model the Bayesian filter is able to track the system's belief over time. The recursive Bayesian filter equation adapted to parameter estimation is defined by:

$$Bel(\mathbf{\Theta}_N) = \eta \, p(\mathbf{Z} = \mathbf{z}_N|\mathbf{\Theta}_N) \int_{\mathbf{\Theta}_{N-1}} p(\mathbf{\Theta}_N|\mathbf{\Theta}_{N-1}) Bel(\mathbf{\Theta}_{N-1}) \, d\mathbf{\Theta}_{N-1} \ , \quad (1)$$

where $p(\mathbf{Z} = \mathbf{z}_N|\mathbf{\Theta}_N)$ denotes the observation model, $p(\mathbf{\Theta}_N|\mathbf{\Theta}_{N-1})$ denotes the transition model and $Bel(\mathbf{\Theta}_{N-1})$ is the prior belief on the unknown pa-

rameters. The left term of the equation describes the posterior probability and $\eta = p(\mathbf{Z} = \mathbf{z}_N)$ is a normalization term. Figure 1 shows the parameter estimation on the basis of the Bayesian filter in a factor graph, whereby arrows indicate the messages sent to estimate the posterior belief.

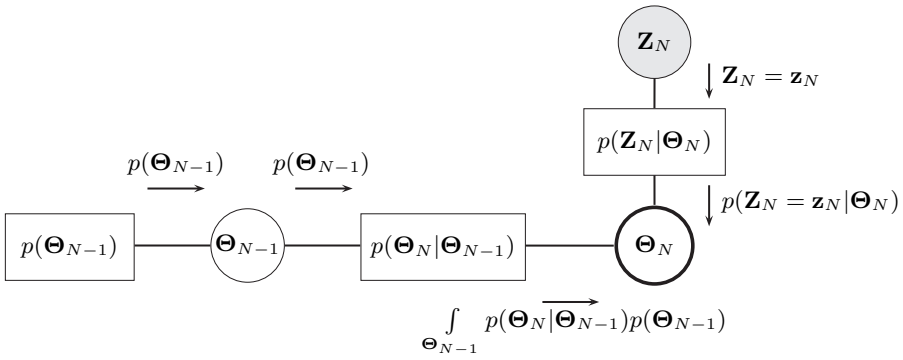

Fig. 1. Parameter estimation with Bayesian filter. The system state $\mathbf{\Theta}$ can be updated by new incoming observations $\mathbf{z}_N$.

When using arbitrary distributions for state representation several problems arise. First, when sending a message from variable node $\mathbf{\Theta}_{N-1}$ to factor node $p(\mathbf{\Theta}_N|\mathbf{\Theta}_{N-1})$ the marginal of the probability distribution $p(\mathbf{\Theta}_{N-1})$ is required. This corresponds to the conjugate prior of the distribution $p(X|Y)$. For arbitrary distributions, modeling the conjugate prior is very hard [9]. Another problem is the design of the conditional probability distribution $p(\mathbf{\Theta}_N|\mathbf{\Theta}_{N-1})$ which has to transform the hyperparameters into a new time step. Additionally, some variables of the state representation can depend on unobserved variables.

3 MAP Estimation of Dynamically Changing Distributions

This section presents an extended factor graph structure and a loopy belief propagation algorithm with augmented message types, that addresses the aforementioned problems. Without loss of generality, we show the state estimation of probability distribution $p(\mathbf{Z}) = p(X|Y)$, where variable Y is hidden.

Figure 2 shows the proposed architecture. We assume a two layered dynamic factor graph, that combines the inference on latent variables in each time step with the estimation of factor potentials over time. The lower layer represents the system's state in its factor potentials and corresponds to a factor graph with conventional algorithms for message passing and inference. The upper layer replaces the transition model and the conjugate prior of the state's probability distribution defined by the factor potential, which is going to develop over time (compare Fig. 1). Thereto, every factor node of the lower layer is linked to a

counterpart in the upper layer, where it acts similar to a variable node in conventional factor graphs. We introduce a new node type denoted by the diamond shape: the hyperfactor node. It acts exactly like a factor node in conventional factor graphs and represents the transition model. Note that we integrated the factor potential $p(Y)$ in distribution $p(X|Y)$ to receive a factor potential $p(X,Y)$ to simplify the example.

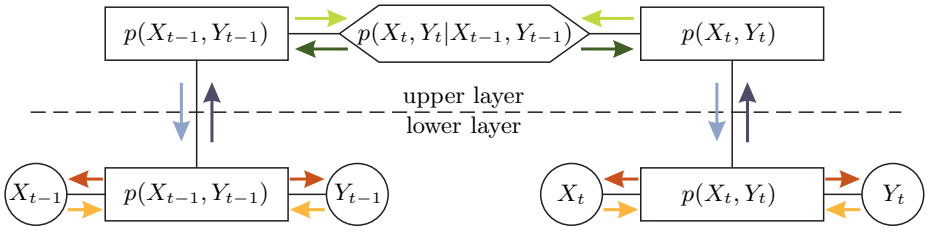

Fig. 2. Extended dynamic factor graph. The figure shows a two layered factor graph, which avoids modeling of conjugate priors and the transition model of the hyperparameters by estimating factor potentials over time. The messages in the proposed architecture are coded in terms of color.

3.1 Message Types

We apply loopy belief propagation for message passing, where each node iteratively calculates and updates all outgoing messages. Not yet calculated messages are supposed to be uniformly distributed.

A message from a variable node V_i to a factor node F_j (yellow arrows in Fig. 2) is defined by common sum product algorithm:

$$\mu_{V_i \to F_j}(\mathbf{X}_{V_i}) = \prod_{F_k \in ne(V_i) \setminus F_j} \mu_{F_k \to V_i}(\mathbf{X}_{V_i}) \ . \tag{2}$$

The variable node V_i multiplies all incoming messages from connected factor nodes $F_k \in ne(V_i)$, except destination node F_j.

A message from a factor node F_i to a variable node V_j (red arrows in Fig. 2) is also defined by common sum product algorithm:

$$\mu_{F_i \to V_j}(\mathbf{X}_{V_j}) = \int_{V_k \in ne(F_i) \setminus V_j} \left(\mu_{F_i^u \to F_i^l}(\mathbf{X}_{F_i}) \prod_{V_k \in ne(F_i) \setminus V_j} \mu_{V_k \to F_i}(\mathbf{X}_{V_k}) \right) dV_k \ . \tag{3}$$

The factor node F_i multiplies all incoming messages from connected variable nodes $V_k \in ne(F_i)$, except destination node V_j with its factor potential. The factor potential complies with the message from the upper layer $\mu_{F_i^u \to F_i^l}(\mathbf{X}_{F_i})$ defined in (5). The result is marginalized on the correct type of the destination node by integrating over all other variables.

The factor potentials should be transferred in the new state by applying the transition model in the hyperfactor nodes. Therefore, the factor potential of a factor node F_i in the upper layer has to incorporate all estimations of the parameters and the current observation. The estimation of the factor potentials $p(X_{i-1}, Y_{i-1})$ and $p(X_{i+1}, Y_{i+1})$ comprise the estimation of the system's state given observations $x_1, \ldots, x_{i-1}$ of previous time steps and observations $x_{i+1}, \ldots, x_N$ of future time steps, respectively. The knowledge of the lower layer $p(X = x_i, Y)$ corresponds to the current observation x_i.

For that purpose, a factor node F_i^l of the lower layer has to calculate the joint of the marginal of its local neighborhood given the current observation and sends it to the factor node F_i^u in the upper layer (dark blue arrows in Fig. 2):

$$\mu_{F_i^l \to F_i^u}(\mathbf{X}_{F_i}) = \prod_{V_k \in ne(F_i^l)} \mu_{V_k \to F_i^l}(\mathbf{X}_{V_k}) \prod_{V_k \in ne(F_i^l)} \mu_{F_i^l \to V_k}(\mathbf{X}_{V_k}) \ . \tag{4}$$

The factor node multiplies all incoming messages $\mu_{V_k \to F_i^l}(\mathbf{X}_{V_k})$ from connected variable nodes $V_k \in ne(F_i^l)$ and all outgoing messages $\mu_{F_i^l \to V_k}(\mathbf{X}_{V_k})$ to these nodes. The incoming messages from connected variable nodes correspond to messages, that the variable nodes received themself from possible existing other factor nodes in the lower layer and possible observations. To receive the correct marginals of the variable nodes in the local neighborhood, the messages from the factor node F_i^l to the variable nodes are multiplied.

The factor node of the lower layer should use the estimation from all time steps, except the current, to estimate its potential. Therefore, a message from the upper layer to the lower layer (light blue arrows in Fig. 2) results in:

$$\mu_{F_i^u \to F_i^l}(\mathbf{X}_{F_i}) = \prod_{H_k \in ne(F_i^u)} \mu_{H_k \to F_i^u}(\mathbf{X}_{F_i}) \ . \tag{5}$$

The factor node multiplies all incoming messages from all connected hyperfactor nodes $H_k \in ne(F_i^u)$.

The algorithm to calculate a message from a hyperfactor node H_i to a factor node F_j (dark green arrows in Fig. 2) is exactly the same as a message of a factor node in traditional sum product algorithm:

$$\mu_{H_i \to F_j}(\mathbf{X}_{F_j}) = \int_{F_k \in ne(H_i) \setminus F_j} \left(p\left(ne\left(H_i\right)\right) \prod_{F_k \in ne(H_i) \setminus F_j} \mu_{F_k \to H_i}(\mathbf{X}_{F_k}) \right) dF_k \ . \tag{6}$$

The hyperfactor node multiplies all incoming messages from connected factor nodes $F_k \in ne(H_i)$, except destination node F_j with its potential $p(ne(H_i))$. The result is marginalized on the correct domain of the destination node.

Finally, we have to define a message from a factor node F_i^u in the upper layer to a hyperfactor node H_j (light green arrows in Fig. 2):

$$\mu_{F_i^u \to H_j}(\mathbf{X}_{F_i}) = \left(\prod_{H_k \in ne(F_i^u) \setminus H_j} \mu_{H_k \to F_i^u}(\mathbf{X}_{F_i}) \right) \oplus \mu_{F_i^l \to F_i^u}(\mathbf{X}_{F_i}) \ . \tag{7}$$

The factor node F_i^u multiplies all messages from connected hyperfactor nodes $H_k \in ne(F_i^u)$, except the destination H_j. This corresponds to the previous and future estimations. The message from the lower layer $\mu_{F_i^l \to F_i^u}(\mathbf{X}_{F_i})$ corresponds to the current observation. It is added by a MAP estimate step denoted by operator $\oplus$. The realization of the MAP estimation function depends on the type of distribution used within the factor potentials. The estimate step is possible even with hidden parameters, because the algorithm provides an estimation for these parameters. This concept is closely related to EM.

3.2 Relation to Expectation Maximization

EM finds maximum likelihood solutions for models having latent variables. Given a joint distribution $p(X, Y|\mathbf{\Theta})$, where variable Y is latent, the goal is to maximize $p(X|\mathbf{\Theta})$ with respect to $\mathbf{\Theta}$ [2]. Figure 3 shows the E step of EM interpreted as an inference problem in a factor graph. If one assumes a prior on the parameters $\mathbf{\Theta}$, the marginal $p(Y|X, \mathbf{\Theta})$ can be inferred given all observations $X = \{x_1 \dots x_N\}$ by message passing. The factor graph in Fig. 3 matches the lower layer of the extended factor graph presented in Fig. 2. Therefore, the calculation of the marginals and the message from the lower to the upper layer defined in (4) correspond to the expectation of an observation.

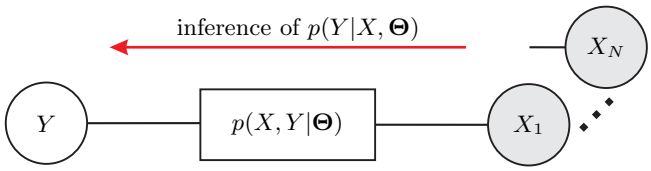

Fig. 3. Expectation step in a factor graph. The inference of marginal $p(Y|X, \mathbf{\Theta})$ corresponds to the E step of EM algorithm.

With the inferred marginal $p(Y|X, \mathbf{\Theta})$, all parameters of the probability distribution can be treated as being observed, and the maximization step follows. The M step evaluates $\mathbf{\Theta}'$ with:

$$\mathbf{\Theta}' = \arg\max_{\mathbf{\Theta}} \left(\sum_{Y} p(Y|X, \mathbf{\Theta}) \ln p(X, Y|\mathbf{\Theta}) \right) \tag{8}$$

The two steps are iterated successively to update the estimation of the model parameters. In the proposed architecture the M step is replaced by a MAP estimation step, which integrates the message of the lower layer into the factor potential in each time step in (7). Therefore, the belief of the state – represented by the parameters of the factor potential – is updated sequentially by splitting up the sums of the arg max operation in (8) into local operations in each time step of a dynamic factor graph. The MAP estimation step of discrete distributions

using conjugate priors is equivalent to a weighted sum of the distributions [2]. For $\lim_{\Delta x \to 0}$ we can transfer this observation to arbitrary distributions. Therefore, in case of hardly manageable conjugate priors of arbitrary distributions we suggest to approximate the MAP estimation step (operation $\oplus$ in (7)) by a weighted sum. In case of manageable conjugate prior distributions on hyperparameters a regular MAP estimation step can be applied instead. The iterations of the loopy belief propagation correspond to the iterations of EM algorithm. Additionally, the hyperfactor node in the upper layer introduces a transition model, that has to be considered in the estimation process.

3.3 Transition Model

Generating the transition model is very intuitive, because it acts directly on the variables of the factor potentials instead of the hyperparameters of the model. For convenience, we assume Gaussians for variable $\mathbf{X}$ with $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. One receives similar results by using discrete or mixed distributions [2]. When using an identical transition model the common MAP estimation can be utilized to add the current observation to the estimation in the upper layer like in (7).

After N observations, there should exist estimations for the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with $\hat{\boldsymbol{\mu}}_{\boldsymbol{N}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{N}}$. The parameter of the Gaussian after the integration of the new observation $\mathbf{X} = \mathbf{x}_{N+1}$ result in:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{N+1} &= \frac{N}{N+1}\, \hat{\boldsymbol{\mu}}_N + (1 - \frac{N}{N+1})\, \mathbf{x}_{N+1} \\ \hat{\boldsymbol{\Sigma}}_{N+1} &= \frac{N}{N+1}\, \hat{\boldsymbol{\Sigma}}_N + \frac{N}{(N+1)^2} (\hat{\boldsymbol{\mu}}_{N+1} - \mathbf{x}_{N+1})(\hat{\boldsymbol{\mu}}_{N+1} - \mathbf{x}_{N+1})^T \quad . \end{aligned} \tag{9}$$

Intuitively, the new observation $\mathbf{x}_{N+1}$ is added to the estimation and weighted. The weight accounts for the prior knowledge about the location and form of the distribution given the previous observations. We introduce an additional weighting factor $\alpha = [0, 1]$ to weaken the influence of the former observations of the MAP estimation step in the upper layer in (7). Thus the term $\frac{N}{N+1}$ of (9) is weighted as follows:

$$\frac{\alpha N}{\alpha N + 1} \quad . \tag{10}$$

The weighting factor α is set to 1 if the transition model is identical:

$$p(\boldsymbol{\Theta}_N | \boldsymbol{\Theta}_{N-1}) = \begin{cases} 1 & , \text{if } \boldsymbol{\Theta}_N = \boldsymbol{\Theta}_{N-1} \\ 0 & , \text{else} \end{cases} \tag{11}$$

Once the transition model is non identical, it blurs the unknown parameters of the distribution in the hyperfactor nodes of every time step. Hence, the weight of former observations has to be adopted by decreasing α. The weighting factor is calculated empirically depending on the uncertainty of the transition model. Therefore, α constitutes the difference between keeping previous learned knownledge or integrate new knowledge into the model given by an observation.

4 Discussion and Conclusion

The used loopy belief propagation only provides an approximation of the true marginals. Additionally, the convergence of loopy belief propagation is not proven for arbitrary graph structures. Because the concept is closely related to EM, it is not guaranteed that the algorithm converges to the global maximum. The weighting of the current observation in (10) when using non identical transition models is hardly possible if the upper layer of the extended factor graph contains loops. This is primarily induced by circulating messages in the upper layer, which make it nearly impossible to find an adequate value of α. The algorithm works for arbitrary distributions as long as the MAP estimation operation of one observation to the distribution is defined or a weighted sum can be approximated. Last but not least, the algorithm is only real time capable for small graphs with few nodes. Otherwise, loopy belief propagation algorithm needs many computational expensive iterations to converge.

This paper presented a concept to allow handling of online state estimation represented by dynamic probability distributions without the need for conjugate priors or hyperparameter transition models. The proposed approximative method extends loopy belief propagation in factor graphs. The key idea is to substitute the ML in the M step of EM by a MAP estimation, which can be applied recursively. The E step is done locally and the iterations of the EM algorithm are shifted into the iterations of loopy belief propagation. Despite of MAP estimators that do not work with hidden parameters or EM algorithm that needs the complete data set of observations, our algorithm offers the possibility to track complex non-stationary system states over time, even if they depend on hidden parameters.

References

1. Thrun, S., Burgard, W., Fox D.: Probabilistic Robotics. MIT Press, Cambridge, MA (2005)
2. Bishop, Christopher M.: Pattern Recognition and Machine Learning. Springer Science+Business Media, Secaucus, NJ, USA (2006)
3. Lauritzen, S.L.: Graphical Models. Clarendon Press, Oxford, UK (1996)
4. Jordan, M.I., Sejnowski, T.J.: Graphical Models: Foundations of Neural Computation. MIT Press, Cambridge, MA (2001)
5. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. UC Berkeley (2002)
6. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47, 498–519 (2001)
7. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, San Mateo, CA (1988)
8. Frey, B.J.,Kschischang, F.R.,Loeliger, H.A., Wiberg, N.: Factor graphs and algorithms. Proceedings of the 35th ACOCCC 1998, 666–680 (1997)
9. Diaconis, P., Ylvisaker, D.: Conjugate priors for exponential families. Annals of Statistics 7, 269–281 (1979)
10. Dempster, A.P., Laird, N.M., Rubin, D.B., others.: Maximum likelihood from incomplete data via the EM algorithm. JRSS 39, 1–38 (1977)