Dirichlet process–based component detection in state-space models

Botond A. Bócsi, **Lehel Csató**

Department of Mathematics and Computer Science
Babeş-Bolyai University
Kogalniceanu str. 1, RO-400084, Cluj-Napoca - Romania

December 11, 2011

Abstract

We present an extension of the switching-state models (SSSM) that allows arbitrary number of linear components in the system. We propose a Bayesian setting where we use Dirichlet process prior over the mixture components of the linear models. This prior allows the inference on the number of linear models in the mixture. We also develop a distance measure in the space of linear Kalman filters with the use of the Kullback-Leibler divergence over the state evolution induced by the individual Kalman filters. The introduced distance measure allows to remove components that are no longer relevant, making the algorithm more effective. We test the proposed algorithm on both artificial and real-world data.

1 Introduction

The analysis of sequential data is an important research topic: this type of data is found in several domains like the analysis of medical data [Baldi and Brunak, 1998], the forecasting of economical fluctuations [Michael P. Clements, 1998, 2001], or in robotics [Yarling, 1992], where the information to be processed lies in sequential data obtained from the sensors [Grewal and Andrews, 2001].

Our goal is to extend the linear Kalman filtering (KF) scheme [Haykin, 2001; Grewal and Andrews, 2001] to a nonlinear framework that is better suited for the usually nonlinear real world data. Although non-linear modelling is a better choice, it is very often computationally infeasible. A possible solution to the intractability is to use a non-linear model built from *locally linear* models [Ghahramani and Hinton, 2000; Murphy, 1998], similar to the mixture models in clustering [Bishop, 2006], called switching-state space models (SSSM).

The idea of using SSSM's for sequential data modelling is not recent. A wide literature is available, it was mainly used in economics and signal processing [Chang and Athans, 1978; Hamilton, 1989; Shumway and Stoffer, 1992; Hill, 1994]. Early researches focused on single valued latent states, models which allow multiple real-valued state vectors were introduced by Ghahramani and Hinton [2000]. It contains the derivation of the learning algorithm for the model parameters as well.

SSSM's are hierarchical models. On the bottom of the hierarchy are simple – linear – models, usually KF's, whereas the interaction among these linear components are governed by Hidden

Markov Models. Due to this hierarchical construction the optimization of the parameters of the SSSM's are rather difficult, sufficient optimization algorithms have not been developed.

We extend the SSSM's with the possibility to determine the number of components in the data–set. This is achieved by using a Dirichlet prior assumption on the structure of the model that allows the insertion of a new local KF and to fit it to the data we have. As the number of components can grow indefinitely, we need a mechanism that *trims* the model by *removing* filters that are not used. This is achieved with the introduction of a distance measure in the space of the local Kalman filters.

It is an important consequence of the proposed algorithm that it can be used in a setting where the component detection is independent from the controlling mechanism of the robot, allowing thus the identification of *motion* primitives that might form the basis of an ontology within the space of internal representation of the robot [Houser and Kloesel, 1992].

The paper is organised as follows: first an introduction into the topic of SSSM's is given and the Dirichlet prior on the structure of the model is introduced. Section 3 contains the proposed parameter estimation algorithm for the SSSM's. The new Kalman filters measure is described in Section 4. The simulations we conducted to support our approach are presented in Section 5, with conclusions drawn in Section 6.

2 Switching state-space models

In this paper we build a generative model for the observed data assuming that the data are coming from several distinct sources. We further assume that these sources are "simple", although in general they can also be arbitrarily complex. Here we focus on simple individual models without knowing from which component a specific data was obtained, known as mixture models in machine learning [Ghahramani and Hinton, 2000; Murphy, 1998]. In a general way a SSSM can be formulated as follows:

$$\begin{aligned} x_k^{(s)} &= f^{(s)}(x_{k-1}^{(s)}, w_k^{(s)}) \\ z_k &= h^{(s)}(x_k^{(s)}, v_k^{(s)}), \end{aligned}$$

where $f()$ and $h()$ are arbitrary functions, and the components w_k and v_k are arbitrary noise processes. The superscript $^{(s)}$ – where s is the *switching variable* – defines which component produced the actual z_k output. If we assume N components then $s \in \{1, \ldots, N\}$. The outputs depend on unobserved, latent variables donated by x_k.

At any given time k we observe the value z_k, without knowing which components – defined by s – has generated it. The major problem is to determine the generative component for every observed datum.

Note that if we allow arbitrary complex functions for $f()$ and $h()$ the presence of the mixture model is pointless, because a single function for $f()$ and $h()$ – which are complex enough – can generate all the data. Therefore an evident assumption is to assume *simple* functions for $f()$ and $h()$.

An important issue is the selection of the switching variable when we know the model. We assume that the dynamics of the switching variable s is determined by a Hidden Markov Model (HMM) – we are going to discuss HMM's in Section 2.2.

By using SSSM's we build a generative model in which data come from multiple sources. Although these sources can be arbitrary complex (linear/non-linear, noisy/noiseless), for a better tractability usually it is assumed they are Kalman filters – KF's will be discussed in Section 2.1. The non-linear extension has been studied by Honkela [2001], using neural networks to model

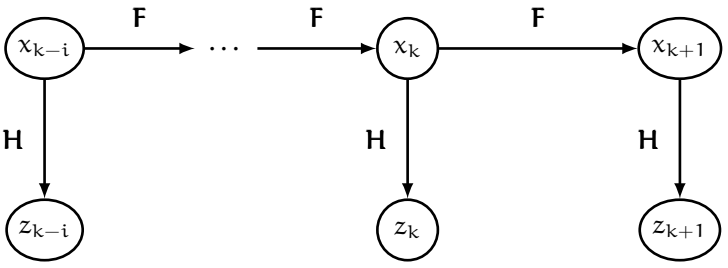


Figure 1: Graphical model of the Kalman filter.

$f()$ and $h()$ respectively. In this paper we do not deal with non-linear components, instead we address the problem of SSSM's, when the sources are Kalman filters. In this case the SSSM is called a switching state Kalman filter (SSKF).

The SSKF framework can be viewed as an extension of the Kalman filters where multiple sources are available, but is also can be seen as a generalisation of the HMM's where the outputs generated by the latent states are not based on a probability distribution but originate from arbitrary Kalman filters. Next we assume that the individual components are KF's meaning simple models.

2.1 Kalman filters

The Kalman filter is the optimal filter that estimates the state of a linear dynamic system when only noisy measurements are available. Due to analytic tractability we require that both operation and measurement noises to be Gaussian. The system was developed by Kalman [1960] and the firsts applications of the filter was in the NASA Apollo program [see Grewal and Andrews, 2001].

For a brief introduction to the Kalman filter, one should consult Welch and Bishop [1995] and details on the filter are *e.g.* in books by Grewal and Andrews [2001] or Haykin [2001].

Kalman filters are linear estimators using linear state space-model, therefore the functions $f()$ and $h()$ are linear. This simply means that they can be defined by a matrix multiplication – $f() \equiv \mathbf{F}$ and $h() \equiv \mathbf{H}$. As a result the state-space model is the following:

$$x_k = \mathbf{F}_k x_{k-1} + \mathbf{B}_k u_k + w_k \quad (1)$$

$$z_k = \mathbf{H}_k x_k + v_k, \quad (2)$$

where k is the time index, x_k is the internal state of the system, z_k is the measurement data, u_k is the control or external input, $\mathbf{F}_k$ is the time-transition matrix, $\mathbf{H}_k$ is the connection between the measurements z_k and the latent states x_k. If we assume that we can interfere with the system, then we can do it using the control input: its effect is linear and characterised by the matrix $\mathbf{B}_k$. There are two noise processes in the dynamical system: the operation noise w_k and the measurement noise v_k, characterised by their respective covariance matrices $\mathbf{R}$ and $\mathbf{Q}$. These have to be *additive* and *white*. Another restriction of eq. (1) is that the noise must be Gaussian with zero mean. A graphical representation of the Kalman filters can be found in Figure 1.

For convenience we assume that the system is stationary so the k index can be omitted in matrices $\mathbf{F}_k$ and $\mathbf{H}_k$. For simplicity we also neglect the control input. We define the Kalman filter by its parameters:

$$\mathrm{KF} = \{\mathbf{F}, \mathbf{H}, \mathbf{Q}, \mathbf{R}\}, \quad (3)$$

The operation of the filter is split into two steps:

1 the **prediction** step, based solely on the previous estimated latent state x_{k-1}:

$$\begin{aligned} x_k^- &= \mathbf{F}x_{k-1}^+ \\ \mathbf{P}_k^- &= \mathbf{F}\mathbf{P}_{k-1}^+\mathbf{F}^T + \mathbf{Q}, \end{aligned} \tag{4}$$

where the first line is the most probable estimation of the latent state given the previous estimation x_{k-1}^+. The second line tells us the *covariance matrix* of the new latent state given the covariance of the previous state $\mathbf{P}_{k-1}^+$ and the known covariance matrix of the operation noise $\mathbf{Q}$.

2 the **update** step, when the new measurement data z_k are included into the estimation process:

$$x_k^+ = x_k^- + \mathbf{K}_k\left(z_k - \mathbf{H}x_k^-\right) \tag{5}$$

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)\mathbf{P}_k^- \tag{6}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R})^{-1}, \tag{7}$$

where x_k^+ is the estimation based on the system equations from the prediction step eq. (4) and the actual observations z_k. The matrix $\mathbf{R}$ is the covariance matrix corresponding to the measurement process. The matrix $\mathbf{K}_k$ is called the *Kalman gain*, it controls the effect that the current measurement has over the new state.

The Kalman filter framework can be used not solely for filtering tasks but also for **smoothing**. A smoother estimates the latent states of the system in time k based on all available measurements before and after the respective moment. For instance, it can be used when we want to estimate the value of a latent state in a time when no measurements were available.

The equations for the Kalman smoother looks as the followings:

$$\begin{aligned} \hat{x}_k &= x_k^+ + \mathbf{J}_k(\hat{x}_{k+1} - \mathbf{F}x_k^+) \\ \hat{\mathbf{P}}_k &= \mathbf{P}_k^+ + \mathbf{J}_k(\hat{\mathbf{P}}_{k+1} - \mathbf{P}_k^-)\mathbf{J}_k^T, \qquad \text{where} \\ \mathbf{J}_k &= \mathbf{P}_k^+\mathbf{F}^T(\mathbf{P}_k^-)^{-1}. \end{aligned}$$

The smoothing process is a backward recursion, the predicting phase starts from the last moment when measurements were observed and goes back until it is needed (usually until the moment of the first observation). The smoothing process always needs to be preceded by a filtering step (forward step). The notion of Kalman smoother always includes both the forward and backward steps.

Note that the adjustments made in the predicted values does not depend directly on the measurement data, only on the estimated values of the forward step. Details can be found in [Yu et al., 2004], [Bishop, 2006], [Grewal and Andrews, 2001].

2.2 Hidden Markov Models

The Hidden Markov Model (HMM) is a widely used method to model discrete sequential data, prevalently used in speech recognition [see Rabiner, 1989], in natural language modelling [see Jurafsky and Martin, 2008], in fault detection [Smyth et al., 1997] or in processing of biological data [see Baldi and Brunak, 1998].

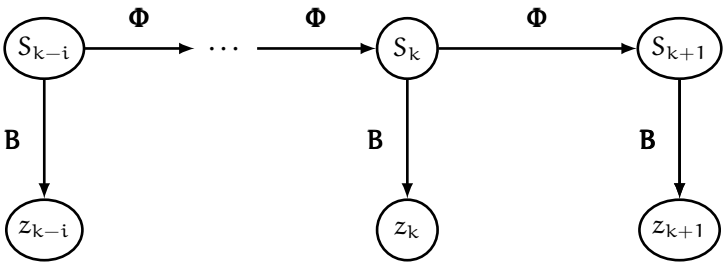


Figure 2: Graphical model of the HMM.

HMM's can be used when relevant values of the system cannot be observed directly and only indirect measurements are available. These measurements have functional relation with the hidden states but are considered independent among each other. The hidden values have to be discrete and are called the hidden states.

The HMM is defined by the following parameters:

$$\text{HMM} = \{\boldsymbol{\Phi}, \mathbf{B}, \boldsymbol{\pi}, N, M\},$$

where N is the number of hidden states, M is the number of observable values, $\boldsymbol{\Phi}$ is the state transition probability distribution matrix $\boldsymbol{\Phi}_{ij} = p(q_{t+1} = S_j | q_t = S_i)$ – where $S \in \{1, \ldots, N\}$ are the possible states of the system and q_t is the state of the model at time t, $q_t \in \{S_1, \ldots, S_N\}$. $\mathbf{B}$ is the observation symbol probability distribution matrix $\mathbf{B}_{ij} = p(v_i \text{ at } t | q_t = S_j)$ – where v_i defines a possible observation, $v_i \in \{v_1, \ldots, v_M\}$ – and π is the initial state distribution ($\boldsymbol{\pi}_i = p(q_1 = S_i)$). Usually the probability distribution $\boldsymbol{\pi}$ is multinomial given by a matrix, however it can have an arbitrary form *e.g.* Gaussian, mixture of Gaussians, or a neural network For a detailed description of the HMM's consult Rabiner [1989]. The graphical representation of the HMM's can be found on Figure 2.

The HMM framework defines three basic problems:

1 Find the likelihood of a sequence of observations, given the HMM model;

2 Find the most probable hidden state sequence which generated a sequence of observations, given the HMM model is known;

3 Find that the parameters of a HMM model ($\boldsymbol{\Phi}$, $\mathbf{B}$, π), given a sequence of observations and assuming N and M are known.

The problems presented above have well defined and efficient solutions. The first problem has a straightforward solution. The second problem can be solved using the recursive algorithm known as the forward-backward algorithm. It is completely analogous to the Kalman smoother (filtering and smoothing) to the discrete case. To learn the parameters of a HMM – third problem – the well-known *Baum-Welch* algorithm can be used. It is based on the EM algorithm using the forward-backward recursion in the E step and the suitable parameter estimation in the M step. All of these algorithms are explained in details by Rabiner [1989]. In this paper we are interested only in the third problem: learning the parameters of a HMM.

2.3 Dirichlet processes

The Dirichlet process (DP) is an extension of the Dirichlet distribution to continuous spaces, therefore a brief introduction of the Dirichlet distribution is given [see Bishop, 2006; Teh, 2007;

Neal, 2000].

The probability density function of the Dirichlet distribution is

$$\mathrm{Dir}(\boldsymbol{\alpha}, \mathbf{u}) = \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma\left(\alpha_k\right)} \prod_k u_k^{\alpha_k - 1},$$

where $\Gamma()$ is the gamma function [see Gradstein and Ryzhik, 1965]. Here $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)^T$ are the parameters of the distribution and u is the set of random variables the distribution is defined on ($0 \leqslant u_k \leqslant 1$ and $\sum_k u_k = 1$). Being defined on random variables, the Dirichlet distribution is probability measure over a probability distribution. As a consequence it is defined on a $k-1$ dimensional simplex. For $k = 2$ we obtain the Beta distribution, so the Dirichlet distribution is a generalisation of the Beta distribution to high-dimensional spaces.

The major advantages of the Dirichlet distribution are apparent in the Bayesian framework, where it can be used as a prior to the multinomial distribution [see Bishop, 2006]. In the non-parametric Bayesian framework it is a suitable choice for a-priori assumptions.

The Dirichlet process (DP) is a continuous extension of the Dirichlet distribution to continuous spaces, when the number of the random variables over which the distribution is defined is infinite.

Before examining the distribution in details we give an example for an experiment that results in Dirichlet process. The most popular experiment is the Pólya urn experiment: suppose we draw balls from an urn containing balls with L different colours. After the draw we replace the ball with two balls having the same colour as the drawn one. But with a probability bigger then zero we put a ball in the urn with a new colour. As the number of the balls tends to infinity, the balls in the urn will be distributed according to a DP. Another experiment is the Chinese restaurant process [see Aldous, 1985; Teh, 2007], it is formulated as follows: there is given a Chinese restaurant with infinite number of tables, each table having an infinite number of seats. The first costumer who arrives seats at the first table. The second costumer either takes a sit beside the first one or chooses a new table. In general the nth costumer chooses a table with a probability proportional to the number of persons sitting at the respective table but there is always the possibility that he/she settles down at a new table. As the number of the costumers tends to infinity, after the previous process their number will be distributed according to a DP. Other experiments can also result in Dirichlet process: e.g. stick-breaking construction, Blackwell-MacQueen urn scheme [see Teh, 2007; Blei et al., 2003].

The rigorous definition of the DP can be formulated as follows: given a $\mathbf{D}$ distribution over a Θ field and the concentration parameter α, $\mathbf{G}$ is distributed according to a Dirichlet process – $\mathbf{G} \sim \mathrm{DP}(\alpha, \mathbf{D})$ – if

$$(\mathbf{G}(A_1), \ldots, \mathbf{G}(A_r)) \sim \mathrm{Dir}\left(\alpha, \mathbf{D}(A_1)\right), \ldots, \mathrm{Dir}\left(\alpha, \mathbf{D}(A_r)\right)$$

for every finite measurable partition $A_1, \ldots, A_r$ of Θ.

The DP is a probability measure over probability measures. The mean of a DP is the base distribution it is defined on – eq. (8) –, whereas the variance is explained by eq. (9).

$$\mathbb{E}[\mathbf{G}] = \mathbf{D} \tag{8}$$

$$\mathrm{Var}[\mathbf{G}] = \mathbf{D}(1 - \mathbf{D})/(\alpha + 1). \tag{9}$$

As one can see the concentration parameter α can be regarded as an inverse variance. As $\alpha \to \infty$, $\mathbf{G} \to \mathbf{D}$. However this convergence can by only weakly or point-wise, because $\mathbf{G}$ is discrete with probability one [see Ferguson, 1973; Teh, 2007]. If the smoothness of $\mathbf{G}$ is required it has to be convolved using kernels [see Teh, 2007].

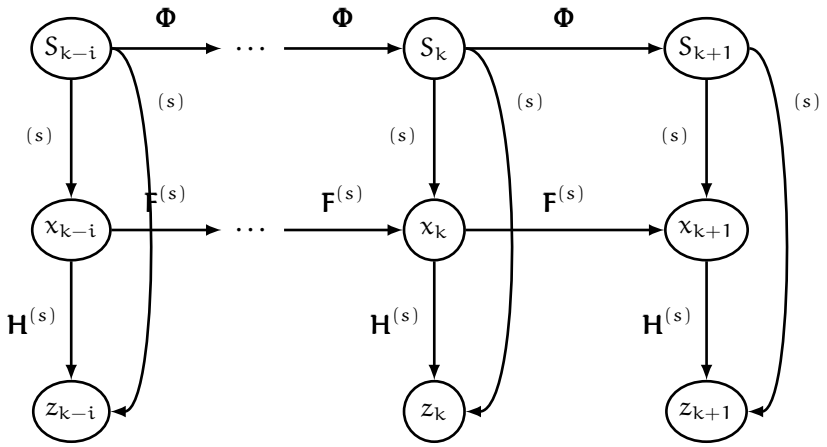


Figure 3: Graphical model of the SSKF.

A crucial task for us is the computation of the posterior distribution of a DP given the observations. The posterior of a DP is intractable therefor alternative methods are advised. Two different approaches are available to perform the evaluation of the posterior. The first one is a mean-field variational method introduced by Blei and Jordan [2004]. The second class of solutions are Monte Carlo Markov Chain – MCMC – based sampling methods. Algorithms were developed both for conjugate priors and non-conjugate priors for the base distribution, using Gibbs sampling or Matropolis-Hasting updates. The main difference among the methods is underlying construction of the DP they assume: mixture of distributions, stick-breaking construction. Details on the MCMC methods are in [Neal, 2000].

As it was shown by Teh [2007]; Neal [2000], the posterior distribution of a DP, given n observed values $\theta_1, \ldots, \theta_n$ – $\mathbf{G}$ being only a distribution on the space Θ – is also a DP with the following parameters:

$$\mathbf{G}(\theta_1, \ldots, \theta_n) \sim \mathrm{DP}\left(\alpha + n, \frac{\alpha}{\alpha + n}\mathbf{D} + \frac{n}{\alpha + n}\frac{\sum_{i=1}^{n} \delta_{\theta_i}}{n}\right),$$

where δ_{θ_i} is the point mass located at θ_i.

In the section we present the special case of the SSSM's when the local models are Kalman filters. The extension of the model using a Dirichlet process a-priori is also presented.

Kalman filters are linear estimators, non-linearity is introduced via the locally linear switching state-space models (SSSM's), where the sources are Kalman filters. A graphical representation of the SSKF can be found in Figure 3. If we denote the observations with z_k and the latent vector as x_k, the state-space equations are as follow:

$$x_k^{(s)} = \mathbf{F}^{(s)} x_{k-1}^{(s)} + w_k^{(s)}$$
$$z_k = \mathbf{H}^{(s)} x_k^{(s)} + v_k^{(s)},$$

where $\mathbf{F}^{(s)}$ is the time transition matrix, $\mathbf{H}^{(s)}$ is the output observation matrix. We assume that the dimensions of the matrices are consistent such that all multiplications can be performed in

eq. (10). The random variables $w_k^{(s)}$ and $v_k^{(s)}$ are the driving and observation noise processes, characterised by respective covariance matrices $\mathbf{R}^{(s)}$ and $\mathbf{Q}^{(s)}$. The superscript $^{(s)}$ – where s is the *switching variable* – defines which component produced the actual z_k output. If we assume N components then $s \in \{1, \ldots, N\}$. The dynamics of s is defined by a Hidden Markov Model – HMM – (see Section 2.2). Let $\mathbf{\Phi}$ be the transition probability and $\boldsymbol{\pi}$ the initial state distribution of the HMM. To be able to estimate the parameters of the local filters, we have to assign the individual data points to a filter, i.e. we have to know the value of s for each z_k, done with the use of the HMM. The complete set of parameters of the SSKF is the following:

$$\begin{aligned} \theta &= \{\theta^{(s)}\}_{s=1}^{N} \cup \{\boldsymbol{\pi}, \mathbf{\Phi}\}, \\ \theta^{(s)} &= \{\mathbf{F}^{(s)}, \mathbf{H}^{(s)}, \mathbf{Q}^{(s)}, \mathbf{R}^{(s)}\}. \end{aligned}$$

We cannot assume we know N a-priori, thus we impose a prior distribution over the components and the number of components our model has. The choice of the Dirichlet prior looks convenient because of its useful properties (e.g. the measures drawn from a Dirichlet process are discrete with probability one):

$$\theta^{(s)} \sim \mathrm{Dir}(\alpha, \mathbf{G}),$$

where $\mathbf{G}$ is the base distribution and α is the concentration parameter – used to set the range of the components of the model. The distribution $\mathbf{G}$ can be relevant regarding the efficiency of the algorithm, however its choice must depend on the nature of the data. Therefore to preserve generality we assumed an uniform distribution on the space of the parameters. Details about the Dirichlet process are presented in Section 2.3.

We mention that with the proposed hierarchical approach we can deal with data build from a potentially infinite number of sources. This is possible since the structure of the model is not fixed and its complexity is dependent on the data. We next describe the inference algorithm.

3 Learning the SSKF parameters

The learning algorithm for the presented mixture model does not have an analytic deduction, we have to use more sophisticated methods to obtain the values of the parameters. Different learning algorithm for SSKF's are discussed by Hamilton [1989]; Shumway and Stoffer [1992]; R.J. Eliott and Moore [1995] but all of these put some constrains on the state space of the model. A general parameter estimation was introduced by Ghahramani and Hinton [2000] and by Murphy [1998]. The inference is a variational method based on a modified version of the Expectation-Maximization (EM) algorithm. We extended this algorithm to infer not only the parameters of the local KF's and the global HMM, but also the number of Kalman filters required by the data.

We do not describe the EM algorithm, detailed presentation can be found in [Dempster et al., 1977; Bishop, 2006]. We also do not treat the problem of HMM parameter estimation – the third problem from Section 2.2 –, because it is a well-known algorithm, well explained by Rabiner [1989] and Bishop [2006].

Learning the parameters of a Kalman filter is also crucial in our framework, we next describe the inference algorithm.

3.1 Learning the Kalman filter parameters

Our goal is to estimate the parameters of a Kalman filters eq. (3), given the observations $z_1, \ldots, z_m$. The inferring algorithm is presented by Bishop [2006], Yu et al. [2004] and Murphy [1998].

The learning method is based on the EM algorithm. Initially the parameters are set to random values and the the following steps are iterate until convergence is reached:

In the **E step** we run the Kalman filter algorithm to assign a probability for each latent state – eq. (6). The expectations on the latent states have to be conditioned on the whole data–set, therefore the Kalman smoother has to be also performed.

In the **M step** we need the values of the following expectations:

$$\begin{aligned} \mathbb{E}[x_k] &= \hat{x}_k \\ \mathbb{E}[x_k x_{k-1}^T] &= \mathbf{J}_{k-1}\hat{\mathbf{P}}_k + \hat{x}_k \hat{x}_{k-1}^T \\ \mathbb{E}[x_k x_k^T] &= \hat{\mathbf{P}}_k + \hat{x}_k \hat{x}_k^T, \end{aligned} \tag{10}$$

where the values of $\hat{x}_k$, $\mathbf{J}_k$ and $\hat{\mathbf{P}}_k$ are obtained from the Kalman smoother – see Section 2.1. We obtain the update equations for the parameters by maximising the complete-data log likelihood function respect to every parameter. Both the driving equation (eq. (1)) and the measurement equation (eq. (2)) are Gaussians, therefore the maximization of the log likelihood is tractable .The complete-data log likelihood has the following form [see Bishop, 2006]:

$$\begin{aligned} \ln p(Z, X|\mathbf{F}, \mathbf{H}, \mathbf{Q}, \mathbf{R}) &= \ln p(x_1|\hat{x}_0, \mathbf{P}_0) + \sum_{i=2}^{M} \ln p(x_i|x_{i-1}, \mathbf{F}, \mathbf{Q}) \\ &+ \sum_{i=2}^{M} \ln p(z_i|x_i, \mathbf{H}, \mathbf{R}). \end{aligned}$$

By maximising the previous equation with respect to the individual parameters and using the expectations from eq. (10), we obtain the following update formulas:

$$\mathbf{F} = \left(\sum_{k=2}^{M} \mathbb{E}[x_k x_{k-1}^T]\right)\left(\sum_{k=2}^{M} \mathbb{E}[x_{k-1} x_{k-1}^T]\right)^{-1} \tag{11}$$

$$\mathbf{Q} = \frac{1}{M-1}\sum_{k=2}^{M}\left\{\mathbb{E}[x_k x_k^T] - \mathbf{F}\mathbb{E}[x_{k-1} x_k^T] - \mathbb{E}[x_{k-1} x_k^T]\mathbf{F} + \mathbf{F}\mathbb{E}[x_{k-1} x_{k-1}^T]\mathbf{F}^T\right\} \tag{12}$$

$$\mathbf{H} = \left(\sum_{k=1}^{M} z_k \mathbb{E}[x_k]\right)\left(\sum_{k=1}^{M} \mathbb{E}[x_k x_k^T]\right)^{-1} \tag{13}$$

$$\mathbf{R} = \frac{1}{M}\sum_{k=1}^{M}\left\{z_k z_k^T - \mathbf{H}\mathbb{E}[x_k] z_k^T - z_k \mathbb{E}[x_k^T]\mathbf{H} + \mathbf{H}\mathbb{E}[x_k x_k^T]\mathbf{H}^T\right\}, \tag{14}$$

where M is the number of observations. Note that applying the update equations for the covariance matrices – eq. 12 and eq. 14 – makes the inferring algorithm highly unstable. To overtake this effect in practical applications we keep their values fixed.

A mostly undiscussed problem of Kalman filter parameter estimation is that it cannot be done unambiguously. Any data–set can be generated by infinitely many Kalman filters. Substituting eq. (1) into eq. (2) and neglecting the control input and the k time index – as reasoned earlier – we obtain

$$z_k = \mathbf{H}\mathbf{F}x_{k-1} + \mathbf{H}w_k + v_k.$$

It can be seen that two degrees of freedom appear between the output z_k and the latent state x_{k-1} – multiplication by $\mathbf{H}$ and $\mathbf{F}$–, therefore one cannot expect to recover the original parameter values of a Kalman filter.

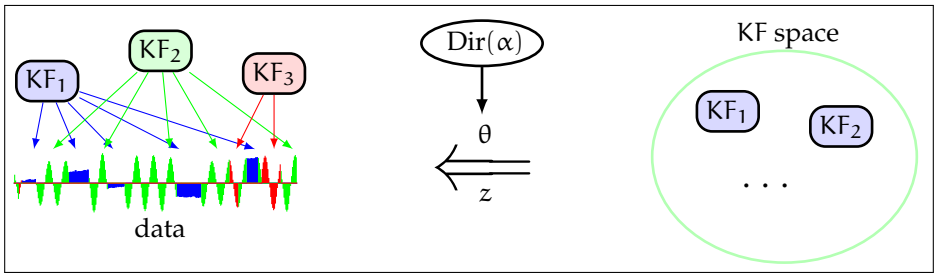


Figure 4: Illustration of the component detection scheme.

3.2 Learning the SSKF parameters with Dirichlet prior

In this Section we describe the learning algorithm for the SSKF's using a Dirichlet prior. We do inference on all the parameters of the individual KF's – $\mathbf{F}, \mathbf{H}, \mathbf{Q}, \mathbf{R}$ – as well as on the parameters of the HMM – $\mathbf{\Phi}, \boldsymbol{\pi}$ –. We also estimate the number of KF's the mixture is composed of using the Dirichlet prior.

Our approach is based on the learning algorithm for SSKF's proposed by Ghahramani and Hinton [2000]. It is a variational learning method using the EM algorithm. The inferring process extended with the Dirichlet priori can be summarise as follows:

[E] In the **expectation** step we do the followings:

E1 we calculate the observation probability – $q_k^{(n)} = p(z_k|S_k = n)$ – for every state-space model from the prediction error of the filters[1];

E2 using $q_k^{(n)}$ as emission probability we calculate the responsibility assigned to every state-space model and every observation – $h_k^{(n)} = p(S_k = n)$ –. This can be easily done by using the forward-backward algorithm for the HMM;

E3 lastly we run the Kalman smoother for every state-space model, using data weighted with the responsibility $h_k^{(n)}$.

[M] In the **maximization** step we do the followings:

M1 we re-estimate the parameters for each Kalman filter – see Section 3.1 –, using data weighted by the responsibility $h_k^{(n)}$ from the E. step;

M2 we re-estimate the parameters of the HMM using the Baum-Welch algorithm [Bishop, 2006; Rabiner, 1989].

[Comp] We also introduce a third step that infers the **number of components**: adds or removes local KF's to the SSKF. First we define the removal procedure then the addition of new models.

A component can be neglected in two cases: either when (1) its contribution drops below a threshold, as suggested by Bishop [2006] with relation to learning the parameters of mixtures models, or (2) when filters generate data very close to each other. The first case is detected by examining the responsibilities $h_k^{(n)}$ from step E. To detect the second case we developed a distance measure between two KF's – defined in Section 4.

[1] Note that in our simulations we used Euclidean distance instead of a Gaussian kernel, proposed by [Ghahramani and Hinton, 2000], because it does not smooth the distances.

The immediate approach that assumes a maximum number of components and during the learning process eliminates unnecessary SSKF's is computationally infeasible, simply because the *maximum number* could be very big. Starting with a small initial component number and increasing its value based on data is a more convenient solution. This can be achieved by using the Dirichlet process extension introduced in Section 2.

We want to get the posterior distribution of the states given by the Dirichlet process. The most convenient way is using Gibbs sampling [Robert and Casella, 2004; Bishop, 2006]. In the Gibbs sampling from a Dirichlet process mixture model [see Neal, 2000; Blei and Jordan, 2004] the following procedure is iterated for every KF model: we sample a KF conditioned on all other switching variables except itself: s_{-n} and the whole data set z:

$$p(s_n^k = 1|z, s_{-n}) \quad \propto \quad p(z_n|z_{-n}, s_{-n}, s_n^k = 1)p(s_n^k = 1|s_{-n}), \tag{15}$$

where the second term can be computed using eq. (16). Let k be the component we have chosen and assume that there are N components. If k is not a new component its probability is given by

$$p(s_n^k = 1|s_{-n}) \quad = \quad \frac{n_k}{\alpha + N - 1},$$

where n_k is the number of occurrences of the k-th filter. The probability that a new filter will be inserted into the SSKF is expressed by the following formula:

$$p(s_n^{N+1} = 1|s_{-n}) \quad = \quad \frac{\alpha}{\alpha + N - 1}. \tag{16}$$

The first term of eq. (15) is the likelihood of the data. It can be obtained using the Kalman smoother for the new potential filter and comparing its likelihood based on the prediction error of the filters.

The Dirichlet process provides means to add new components to the SSKF and we now introduce a method that allows for simplification: we compute a "distance" in the space of Kalman filters presented next.

4 Distance measure of Kalman filters

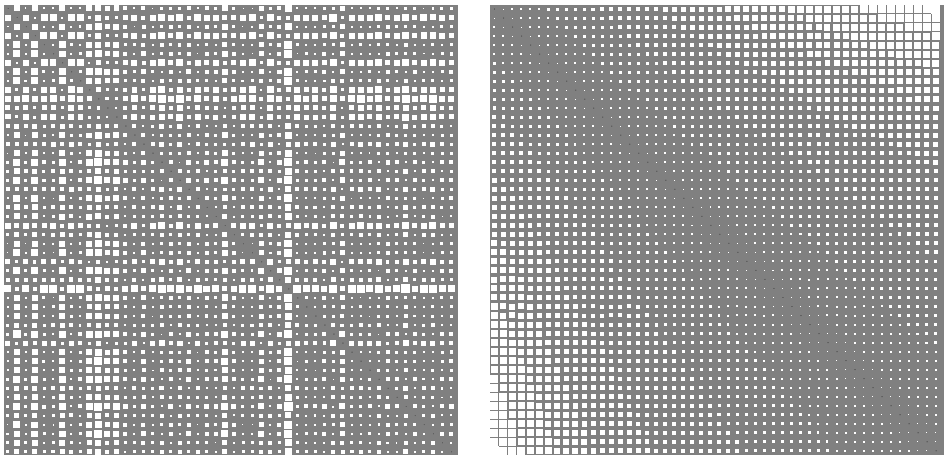
The Kullback-Leibler (KL) divergence is widely used to measure distances between probability distributions. Additionally to its popularity, we know that predictions of the filter are Gaussian random variables and the computation of the KL divergence between Gaussians has analytical form [Kullback, 1959].

We introduce this measure based on KL divergence:

$$d(KF^{(1)}, KF^{(2)}) \stackrel{o}{=} \int dp(z_0)\ KL\left(p(z_1^{(1)}|z_0) \| p(z_1^{(2)}|z_0)\right). \tag{17}$$

In the equation above we defined $KF \stackrel{o}{=} p(z_1|z_0)$ with z_1 the predicted random variable conditioned on z_0. Since z_0 itself is unknown, we treat it again as a random variable $p(z_0) = \mathcal{N}(0, \mathbf{\Sigma}_{z_0})$ and average with respect to it, as shown above. $KF^{(1)}$ and $KF^{(2)}$ are the two Kalman filters and $KL(\cdot, \cdot)$ is the symmetric extension of the KL divergence [Luc Devroye, 1996] between the two conditional predictive distributions. The applied symmetrised divergence has the following form:

$$KL(p^{(1)} \| p^{(2)}) = KL_{reg}(p^{(1)} \| p^{(2)}) + KL_{reg}(p^{(2)} \| p^{(1)}),$$



(a) (b)

Figure 5: Hinton diagrams for (a) random Kalman filters and (b) linearly modified Kalman filters.

where $\mathrm{KL}_{reg}(p^{(1)}\|p^{(2)}) = \int dx\ p^{(1)}(x)\log\frac{p^{(1)}(x)}{p^{(2)}(x)}$ is the KL divergence. Note that other symmetric extensions are also possible [see Luc Devroye, 1996; Johnson and Sinanović, 2001], however this simple form satisfies our claims.

Thus the evaluation of $p(z_1|z_0)$ is needed for each filter, the formula from eq. (18) is obtained in Appendix A. For each KF it looks as follows:

$$p(z_1|z_0) = \mathcal{N}(\mathbf{H}\mathbf{F}\mu_0, \mathbf{H}(\mathbf{F}\mathbf{\Sigma}_0\mathbf{F}^T + \mathbf{Q})\mathbf{H}^T + \mathbf{R}), \tag{18}$$

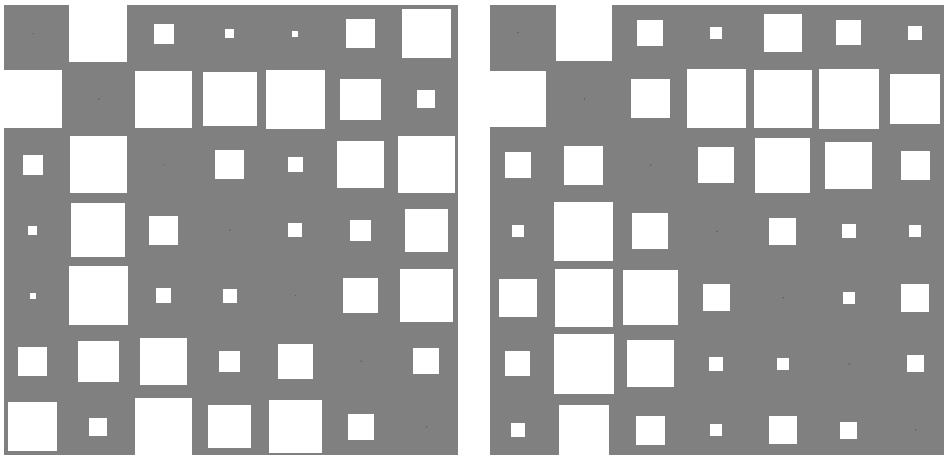
where $\mu_0^T = z_0^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_0^{-1})^{-1}$ and $\mathbf{\Sigma}_0 = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_0^{-1})^{-1}$, where $\mathbf{P}_0$ satisfies $\mathbf{H}\mathbf{P}_0\mathbf{H}^T = (\mathbf{\Sigma}_{z_0} - \mathbf{R})$.

Due to the computability of Gaussian expectations we have an easily obtainable measure between two KF's using minimalist assumptions about the initial value z_0.

In the following part of this section we take a closer look at the new Kalman filter distance. Figure 5 contains two Hinton diagrams, which show the distances of KF's. Both diagrams represents the similarities among 50 filters. Figure 5.(a) contains KF's with random parameters, whereas in Figure 5.(b) every filter is a slightly modified version of the one it precedes (first being the top-left one). As one can see the diagonals of both diagrams are dark, which means that the distance between the same filters is always zero. Figure 5.(b) also reveals that a small change in the parameters of a filter produces a small change in the output space, therefore the distance is small as well.

Figure 6 shows the Hinton diagrams between Kalman filters obtained after the algorithm proposed by us had converged. Note that Figure 6 shows the cases where the algorithm detected seven components.

Also interesting to note that this distance does not depend on the dimension of the latent space of the filters, therefore the presented KL-based distance measure allows for direct comparison between filters of arbitrary latent spaces that produce output of the same dimension.



(a) (b)

Figure 6: Hinton diagrams for filters obtained from simulations.

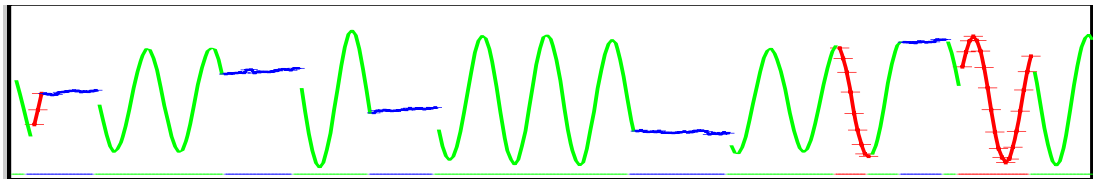


Figure 7: Partitioning of rotation data.

5 Simulations

We tested our method on artificial and real–world data, with our main interest being the inference on the number of components, however we also outline other aspects of the method we introduced. We used three data–sets to examine the effectiveness of our proposal. The first was created using three filters, each with a two dimensional latent space and two dimensional output space. The first filter implemented rotation ($\pi/12$ degree), the second one left the data unchanged, and the third one was also rotation, in opposite direction to the first filter ($-\pi/12$ degree). The observations were corrupted with zero mean normal noise with variance 0.1. We plotted the first dimension on Figure 7, each detected component with a different style. We see that indeed the proposed algorithm identifies mostly correctly the components.

In the second experiment we wanted to partition the three dimensional Lorenz attractor with parameters $\rho = 28$, $\sigma = 10$ and $\beta = 8/3$ [see Strogatz, 2001]. There were just two components left, as it is shown in Figure 8.

The third set is the KIN40 data–set[2], a realistic simulation of the forward dynamics of an 8 link all-revolute robot arm. Figure 9 shows the first 300 points of the first dimension of the data–set, using 4 dimensional latent states. However, discovering partitions in this data–set does not have practical meaning, so the purpose of examining it was to find out whether our algorithm is consistent.

We run all three experiments with different hidden dimension of the internal states and

[2]http://ida.first.fraunhofer.de/ anton/data/kin40k.mat

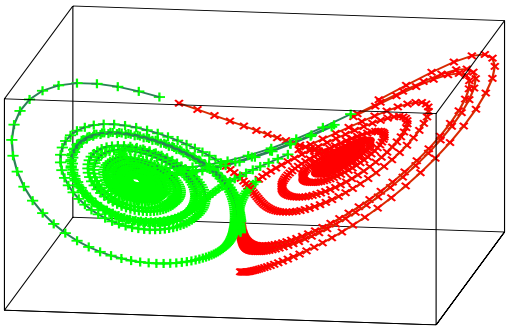


Figure 8: Partitioning of the Lorenz attractor.

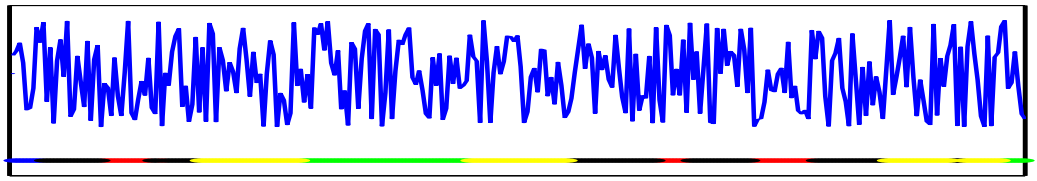


Figure 9: Partitioning of KIN40 data–set.

different prior on the number of sources. Each parameter settings were tested forty times, totally performed 3×600 experiments.

We defined a maximum number of steps – 100 for each simulation – in which the algorithm had to converge. If the algorithm exceeded this limit we assumed that it did not converge. Introducing this limit was necessary because of the well known property of the EM algorithm that it does not always converges, it can stuck in local minima. We also defined a minimum number of steps – 10 for each simulation.

All the figures from this section are composed of two diagramms. The left one always shows the diagram obtained from the simulations with the immediate approach – discussed in Section 2.3 – (we will refer to this method as the *simple* algorithm), whereas the right one contains the results when the DP prior assumption was used. Each diagram shows the results (1) in function of the a-priori assumptions about the number of components (1 to 8). In the simple case this is the *maximum number of components*, while in the DP case it is the value of parameter α. On the other axe (2) the dimension of the latent state (1 to 5) is represented.

We defined six different criteria which formed the base of the simulations. The methods were evaluated from the following aspects: the number of active components detected, the empirical error, the rate of convergence, number of removed components caused by low contribution in data generation, number of removed components caused by similarity between the filters and the speed of convergence.

The most important qualifier of our SSKF extension is the number of active components remained after the algorithm has converged, shown in Figure 10. It reveals that in the case of the DP prior the number of components is less dependent of the prior assumptions, specially of the assumption on the number of the filters (α in the DP case). On the contrary, the number of the filters generating the data depends more on the data itself.

All the results obtained form the simulations were achieved producing nearly the same estimation error, without significant improvement or decline; see Figure 11 for an illustration.

Figure 12 and Figure 13 shows how many components have been removed from the SSKF

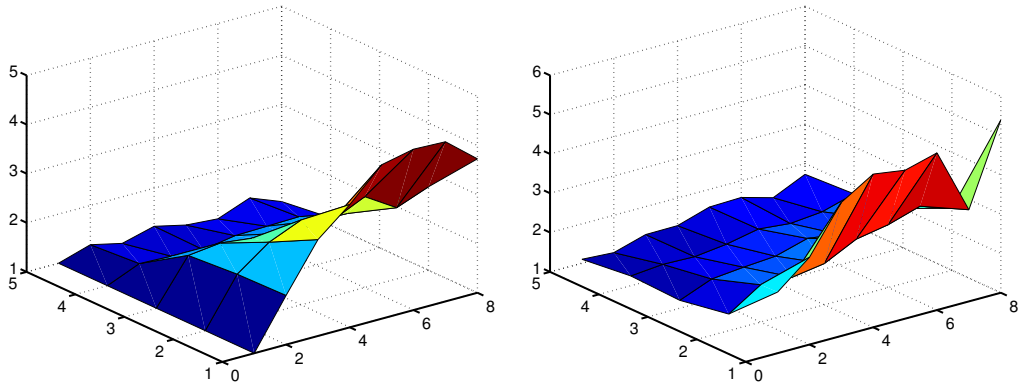


Figure 10: The number of active components after convergence of the algorithm.

model during the processing. Figure 12 shows the number of removed components due to their low contribution to the data generation, whereas Figure 13 contains the number of filters removed due to the small distance between them. As one can see the number of the removed filters – meaning the number of needlessly examined filters – grows with the number of initially estimated values of it, however with a smaller rate in the DP case. Using the DP prior fewer adjustment has to be made, meaning that fewer filters needs to be removed.

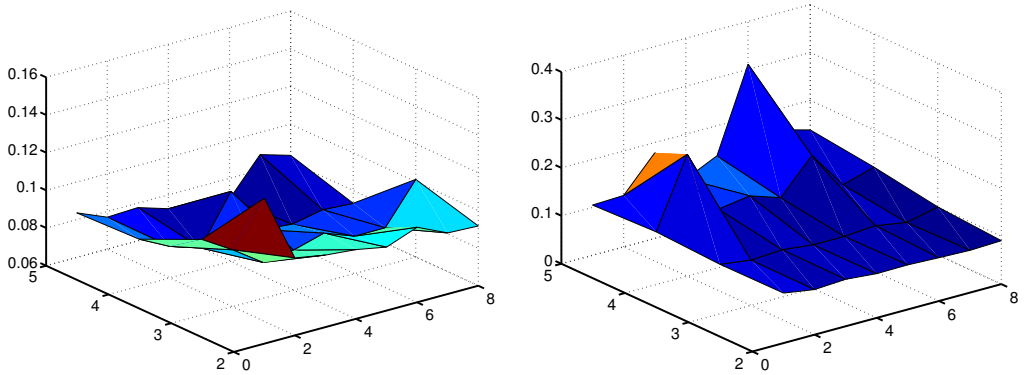
We evaluated the *speed* of the methods using two measures. One consisted of the steps taken until convergence[3] while the other measured the average time – seconds – taken until convergence. As one can see on Figure 14 fewer steps were needed when the DP prior was used mainly because fewer filter has to be examined.

The major advantage of using the DP prior can be seen on Figure 15, which contains the average time – seconds – taken until convergence. The DP solution is more constant does not depend on the prior assumption about the number of the components, e.g. in the simple case supposing 8 component produced unacceptable results, each experiment running over 16 minutes.

6 Discussion

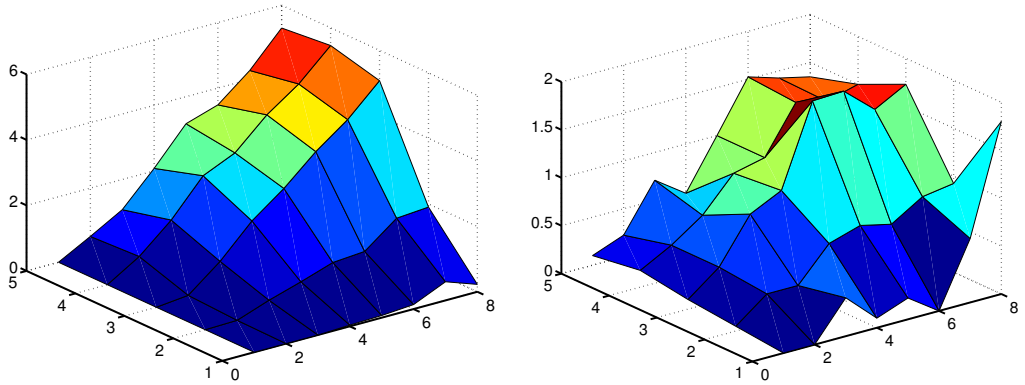
In this paper we presented a generalisation of the SSKF framework by adding a Dirichlet prior over the structure of the model. This extension allows us to model SSKFs whose number of component is not fixed, theoretically can be very high, often infinitely many components and make the computations feasible. We also presented a new distance measure of the Kalman filters, this proved to be useful step in optimising with respect to the number of components of the mixture. The presented simulations show that using our proposal a faster and a more efficient segmentation of the mixture model can be achieved, however, these results rely on generated data. We aim to test the method on real world data too, where the components have practical significance. When locally linear models are insufficient, it is an interesting question is whether there is possible an extension that can deal with models that might involve non-linear filters.

[3]The simple algorithm has not reached convergence in 0.7%, while the DP extension did not converge in 1.8%. The difference is not significant.



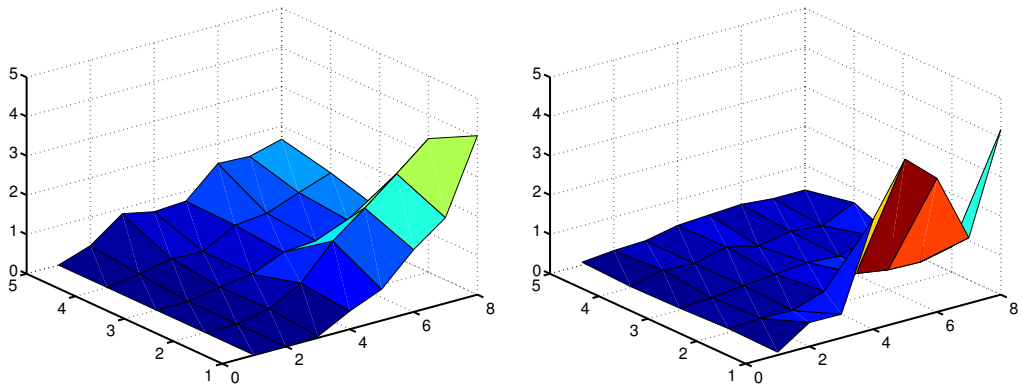
(a) (b)

Figure 11: Empirical error of the estimations.



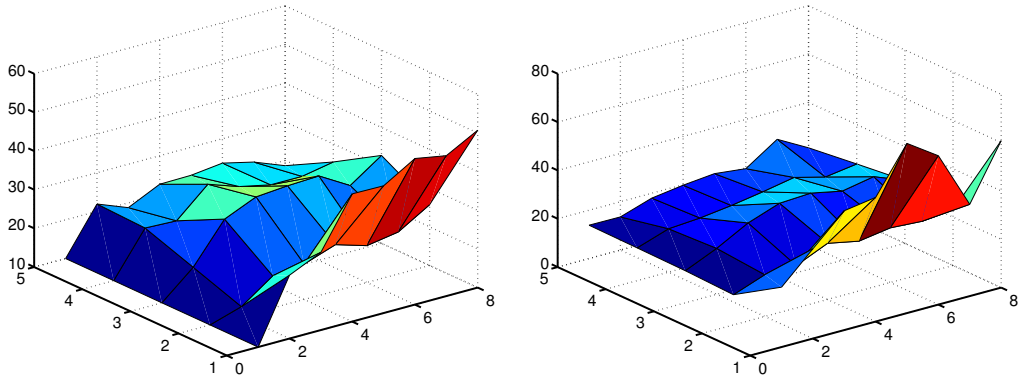
(a) (b)

Figure 12: Number of the removed components caused by their low contribution to data generation.



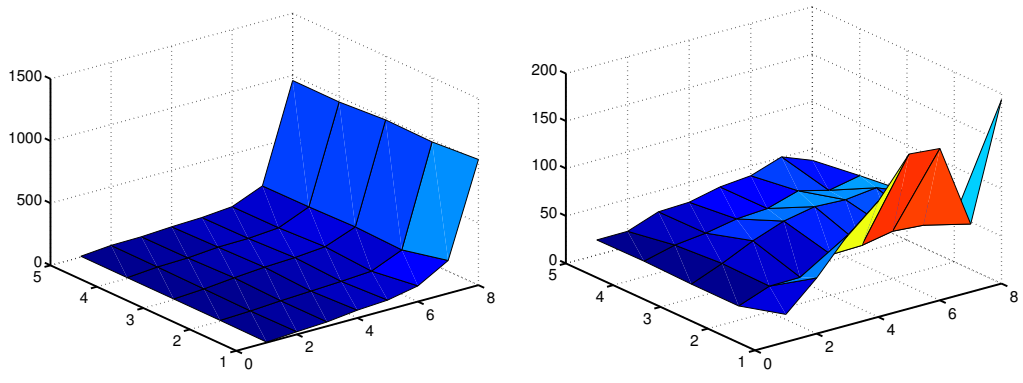
(a) (b)

Figure 13: Number of the removed components caused by similarity between two filters.



(a) (b)

Figure 14: Convergence rate: steps takes until convergence.



(a) (b)

Figure 15: Convergence speed: average seconds until algorithm converges.

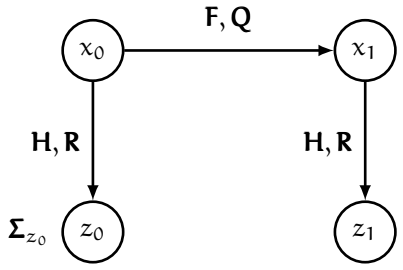


Figure 16: Derivation of the Kalman filter distance.

The presented component detection scheme is *unsupervised*: there is no "user" intervention required for the method to work. Furthermore, the result of this filtering is a set of "simple" models that can be used as building blocks for hierarchical motion planning systems, forming the basis of an ontology. It would be interesting to evaluate the component detection scheme in conjunction e.g. with a reinforcement learning algorithm that can borrow templates from the SSKF.

Acknowledgements

The authors acknowledge the support of the Romanian Ministry of Education, grant PN-II-RU-TE-2011-3-0278.

A Derivation of the Kalman filter distance

The Kalman filter is a linear estimator, therefore if we want to measure the distances between the predicate values of the filters, considering two consecutive values are enough to capture the real *difference* between the filters. Let us define the distance as the following:

$$d(KF^{(1)}, KF^{(2)}) \stackrel{o}{=} \int dp(z_0)\ KL\left(p(z_1^{(1)}|z_0) \| p(z_1^{(2)}|z_0)\right). \tag{19}$$

Using the notation from the previous sections a KF can be defined as $KF \stackrel{o}{=} p(z_1|z_0)$, therefore the first task is to derive the value of $p(z_1|z_0)$, given $p(z_0) = \mathcal{N}(0, \mathbf{\Sigma}_{z_0})$ – see Figure 16:

$$\begin{aligned} p(z_1|z_0) &= \int\int p(z_1, x_1, x_0|z_0)\, dx_1\, dx_0 \\ &= \int\int \frac{p(z_1, z_0, x_1, x_0)}{p(z_0)}\, dx_1\, dx_0 \\ &\propto \int\int p(z_1|x_1)p(x_1|x_0)p(x_0|z_0)\, dx_1\, dx_0 \end{aligned} \tag{20}$$

We start the evaluation posteriorly. Using the Bayes formula the last member of eq. (20) has the following form:

$$
\begin{aligned}
p(x_0|z_0) &= \frac{p(x_0, z_0)}{p(z_0)} = \frac{p(z_0|x_0)p(x_0)}{p(z_0)} \\
&\propto \exp\left\{-\frac{1}{2}(z_0 - \mathbf{H}x_o)^\top \mathbf{R}^{-1}(z_0 - \mathbf{H}x_o)\right\} \exp\left\{-\frac{1}{2}x_0^\top \mathbf{P}_0{}^{-1} x_0\right\} \\
&= \mathcal{N}(z_0^\top \mathbf{R}^{-1}\mathbf{H}(\mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_0^{-1})^{-1}, (\mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_0^{-1})^{-1}) \\
&= \mathcal{N}(\mu_0, \mathbf{\Sigma}_0),
\end{aligned}
$$

where $\mathbf{P}_0$ has to satisfy the following equation:

$$
\mathbf{H}\mathbf{P}_0\mathbf{H}^\top = (\mathbf{\Sigma}_{z_o} - \mathbf{R})\,.
$$

Using the Kalman filter equations the first two terms of eq. (20) can be easily computed:

$$
\begin{aligned}
p(x_1|x_0) &= \mathcal{N}(\mathbf{F}\mu_0, \mathbf{F}\mathbf{\Sigma}_0\mathbf{F}^\top + \mathbf{Q}) \\
p(z_1|x_1) &= \mathcal{N}(\mathbf{H}\mathbf{F}\mu_0, \mathbf{H}(\mathbf{F}\mathbf{\Sigma}_0\mathbf{F}^\top + \mathbf{Q})\mathbf{H}^\top + \mathbf{R}).
\end{aligned}
$$

As one can see none of the values of $p(z_1|x_1)$, $p(x_1|x_0)$ and $p(x_0|z_0)$ depends on x_1 or x_0, so the integrals from the eq.(20) can be omitted, finally obtaining the expression of $p(z_1|z_0)$:

$$
p(z_1|z_0) = \mathcal{N}(\mathbf{H}\mathbf{F}\mu_0, \mathbf{H}(\mathbf{F}\mathbf{\Sigma}_0\mathbf{F}^\top + \mathbf{Q})\mathbf{H}^\top + \mathbf{R}),
$$

where

$$
\begin{aligned}
\mu_0 &= z_0^\top \mathbf{R}^{-1}\mathbf{H}(\mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_0^{-1})^{-1} \\
\mathbf{\Sigma}_0 &= (\mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_0^{-1})^{-1}.
\end{aligned}
$$

Note that the KL distance between Gaussians has closed form, and we have shown that $p(z_1|z_0)$ is always a Gaussian, therefore the evaluation of eq. (19) is straightforward. The KL distance between two Gaussian has the following form [see Kullback, 1959; Cover and Thomas, 1991]:

$$
\begin{aligned}
&\mathrm{KL}(\mathcal{N}(\mu_1, \mathbf{\Sigma}_1), \mathcal{N}(\mu_2, \mathbf{\Sigma}_2)) = \\
&\qquad = (\mu_2 - \mu_1)^\top \mathbf{\Sigma}_2^{-1}(\mu_2 - \mu_1) + \mathrm{tr}(\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-1} - \mathbf{I}) - \ln|\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-1}|.
\end{aligned}
$$

We did not give the exact form of the distance between Kalman filters, instead we have shown that all values are known in order to compute it.

References

D. Aldous. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII 1983*, pages 1–198. Springer, 1985.

P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. The MIT Press, 1998.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D. M. Blei and M. I. Jordan. Variational methods for the dirichlet process. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 12, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5. doi: http://doi.acm.org/10.1145/1015330.1015439.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

C. B. Chang and M. Athans. State estimation for discrete systems with switching parameters. *IEEE Transaction Aerospace Electronic Systems*, AES-14(3):418–424, 1978.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley & Sons, 1991.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Royal statistical Society B*, 39:1–38, 1977.

T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1 (2):209–230, 1973. URL http://www.jstor.org/stable/2958008.

Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4):831–864, 2000.

I. Gradstein and I. Ryzhik. *Table of Integrals, Series and Products*. Academic Press, New York, 1965.

M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. John Wilney and Sons, Inc., second edition, 2001.

J. D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384, 1989. URL http://www.jstor.org/stable/1912559.

S. Haykin. *Kalman Filtering and Neural Networks*. John Wilney and Sons, Inc., 2001.

B. M. Hill. Bayesian forecasting of economic time series. *Econometric Theory*, 10(3-4):483–513, 1994. URL http://ideas.repec.org/a/cup/etheor/v10y1994i3-4p483-513_00.html.

A. Honkela. *Nonlinear Switching State-Space Models*. PhD thesis, Helsinki University of Technology, 2001.

N. Houser and C. Kloesel, editors. *The essential Peirce: selected writings*. Indiana University Press, 1992.

D. H. Johnson and S. Sinanović. Symmetrizing the kullback-leibler distance. Technical report, IEEE Transactions on Information Theory, 2001.

D. Jurafsky and J. H. Martin. *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 2 edition, May 2008. ISBN 0131873210. URL http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0131873210.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

G. L. Luc Devroye, László Györfi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.

D. F. H. Michael P. Clements. *Forecasting Economic Time Series*. Cambridge University Press, 1998.

D. F. H. Michael P. Clements. *Forecasting Non-Stationary Economic Time Series*. MIT Press, 2001.

K. P. Murphy. Switching kalman filters. Technical report, University of California, Berkeley, 1998.

R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

L. Rabiner. A tutorial on hmm and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

L. A. R.J. Eliott and J. Moore. *Hidden Markov Models: Estimation and Control*. Springer-Verlag, New York, 1995.

C. P. Robert and G. Casella. *Monte Carlo Methods*. Springer, second edition, 2004.

R. Shumway and D. Stoffer. Dynamic linear models with switching. *Journal of the American Statistical Association*, (86):763–769, 1992.

P. Smyth, D. Heckerman, and M. Jordan. Probabilistic independence networks for hidden markov probability models. *Neural Computation*, 9:227–269, 1997.

S. H. Strogatz. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry and Engineering*. Perseus Books Group, 2001.

Y. W. Teh. Dirichlet processes. Submitted to Encyclopedia of Machine Learning, 2007.

G. Welch and G. Bishop. An introduction to the Kalman Filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina, 1995.

S. M. Yarling. *A Time Series Modeling Approach for Feedback Control of Robot Arm Positioning Errors*. 1992.

B. M. Yu, K. V. Shenoy, and M. Sahani. Derivation of kalman filtering and smoothing equations. Technical report, Stanford University, 2004.