Support-Vector Conditional Density Estimation for Nonlinear Filtering

Peter Krauthausen[1], **Marco F. Huber**[2], and **Uwe D. Hanebeck**[1]

[1]Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics,
Karlsruhe Institute of Technology, Germany.
[2]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany.
Peter.Krauthausen@kit.edu, Marco.Huber@ieee.org, Uwe.Hanebeck@ieee.org

Abstract – *A non-parametric conditional density estimation algorithm for nonlinear stochastic dynamic systems is proposed. The contributions are a novel support vector regression for estimating conditional densities, modeled by Gaussian mixture densities, and an algorithm based on cross-validation for automatically determining hyper-parameters for the regression. The conditional densities are employed with a modified axis-aligned Gaussian mixture filter. The experimental validation shows the high quality of the conditional densities and good accuracy of the proposed filter.*

Keywords: Nonlinear estimation and filtering, conditional density estimation, support vector regression.

1 Introduction

Estimating the hidden state of a nonlinear stochastic dynamic system lies at the heart of many applications in signal processing, computer vision, robotics, and machine learning. This problem is hard, as analytic, closed-form Bayesian solutions with low, constant complexity cannot be found in general.

There exists a wealth of approximative approaches to nonlinear estimation problems. Regarding the approaches working with certain parameters of the system and measurement functions, the main distinction can be drawn between function approximation approaches, e.g., the *Extended Kalman Filter* (EKF) [1], and density approximation approaches, e.g., the *Unscented Kalman Filter* (UKF) [2]. The former can be understood as linearizing the system and measurement functions. In contrast, the UKF deterministically samples the arising densities and processes these samples using the original nonlinear system and measurement functions.

All of the above approaches use a generative model consisting of nonlinear functions with a fixed noise level. With the advent of Gaussian Processes (GP) [3], sample-based probabilistic models are employed. Thus, for each input one obtains a Gaussian distribution over the output, so as if calculating a set of functions that were Gaussian distributed. Lately, the EKF and UKF approaches were extended to incorporate GP system and measurement relations, yielding the GP-EKF and the GP-UKF [4, 5]. Most interestingly, the explicit function and density approximation performed by the EKF and UKF may be avoided, if one assumes a Gaussian posterior distribution leading to an analytic moment-based GP filter (GP-ADF) [6]. Thus, the GP-ADF allows not only for a processing of samples of a density, but an entire density.

The proposed approach is based on estimating the probabilistic models characterized by nonlinear system and measurement functions, i.e., the conditional densities, by a support vector regression (SVR) [7, 8] from samples only. The resulting probabilistic models in form of mixtures of axis-aligned Gaussians are used with a modified nonlinear Gaussian mixture filter. The obtained posterior distributions are non-Gaussian, allowing for multimodal posterior densities. Note that multimodal densities cannot be supported in any of the above approaches. In addition, this specific type of mixtures allows for closed-form computation of the prediction and filtering step and restricts the complexity to a constant number of components, if the two steps are alternated [9, 10]. Besides the representational benefits, experimental results show that the application of the sample-based conditional density functions delivers good performance at less computational cost: In general, each GP processing step involves iterating over all training samples. For an SVR filter step, one iterates only over all components in the density, which may be significantly less.

The rest of this paper is structured as follows: In Sec. 2, the problem formulation is given. Sec. 3 introduces a novel support vector regression method for estimating the probabilistic models from samples and an algorithm for determining the hyper-parameters. In Sec. 4, a modified axis-aligned Gaussian mixture filter is derived. In Sec. 5, the proposed approach is compared to benchmark density estimators and nonlinear filters.

2 Problem Formulation

In this paper, time-invariant discrete-time stochastic dynamic systems described by a system equation

$$\boldsymbol{x}_k = a(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1} \tag{1}$$

are considered. Here, $a(.)$ denotes a nonlinear mapping of the system state $\boldsymbol{x}_k$, $\boldsymbol{w}_{k-1} \sim \mathcal{N}(0, \sigma_w)$ additive, white Gaussian system noise and k the discrete time index. The relation between $\boldsymbol{x}_k$ and an observation $\boldsymbol{y}_k$ is given by the measurement equation

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \, . \tag{2}$$

Analogously, $h(.)$ denotes a time-invariant nonlinear function and $\boldsymbol{v}_k \sim \mathcal{N}(0, \sigma_v)$ additive, white Gaussian measurement noise. Given the models (1) and (2), a prior distribution $\boldsymbol{x_0}$, and specific measurements $\hat{y}_{1:k}$, the probability density of the hidden state $\boldsymbol{x_k}$ is estimated by recursive filtering—consisting of alternating prediction and filter steps described below.

Prediction Step The density $f(x_k|\hat{y}_{1:k-1})$ is calculated based on the transition density $f(x_k|x_{k-1})$ governed by (1) and the estimate of the hidden state $\boldsymbol{x}_{k-1}$

$$f(x_k|\hat{y}_{1:k-1}) = \int f(x_k|x_{k-1}) \, f(x_{k-1}|\hat{y}_{1:k-1}) \, \mathrm{d}x_{k-1} \, . \tag{3}$$

Filter Step The fusion of the estimate $f(x_k|\hat{y}_{k-1})$ with the latest measurement $\hat{y}_k$ is performed in the filter step, i.e., the information of all measurements up to time step k is combined in this estimate. Using (3) as a prior, the likelihood function $f(y_k|x_k)$ defined by (2) and Bayes' rule, one obtains

$$f(x_k|\hat{y}_{1:k}) = \frac{f(\hat{y}_k|x_k) \, f(x_k|\hat{y}_{1:k-1})}{f(\hat{y}_k|\hat{y}_{1:k-1})} \, . \tag{4}$$

In general, (3) and (4) cannot be solved analytically and may be calculated approximately only. The defining elements are the conditional density functions $f(y_k|x_k)$ and $f(x_k|x_{k-1})$. These conditional densities are determined by (1) and (2) and typically prohibit the closed-form analytic solution to (3) and (4). In some special cases, e.g., conditional densities in the form of Gaussian mixtures, closed-form analytic solutions exist.

In the rest of this paper, the problem of estimating conditional densities in the form of Gaussian mixtures with diagonal covariance matrices, based on i.i.d. data

$$\mathcal{D} = (x_1, y_1), \, \ldots, \, (x_l, y_l) \in \mathbb{R} \times \mathbb{R}$$

is addressed. Thus, it is not required to have the generative models (1) and (2) available in explicit form. Since for many applications, the functions $a(.)$ and $h(.)$ are unknown, estimating the conditional densities based on data only is an essential feature.

3 Conditional Density Estimation

In this section, an algorithm for estimating conditional densities in the form of mixture densities

$$f(y|x) = \sum_{i=1}^{l} \alpha_i \, \mathcal{K}_{\sigma_x}(x, \mu_i^x) \, \mathcal{K}_{\sigma_y}(y, \mu_i^y) \, , \tag{5}$$

with kernel $\mathcal{K}_{\sigma_x}(x, \mu) := \exp\left\{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma_x^2}\right\}$ and weights $\underline{\alpha} = [\alpha_1, \, \ldots, \, \alpha_l]^{\mathrm{T}}$ is proposed. In (5), the kernels are independent for x and y for each mixture component. The mixture consists of one kernel for each data point with the kernels' means being identical to the data points. The estimation consists of the solution of a quadratic optimization problem for determining $\underline{\alpha}$ given fixed hyper-parameters, e.g., kernel widths σ or complexity penalties, embedded in an hyper-parameter optimization.

3.1 Ingredients

For fixed parameters, the estimation algorithm comprises three parts: a term penalizing the density's *complexity*, an *error* term, and the *constraints* necessary to obtain valid conditional density functions. The complexity term

$$\Omega(\underline{\alpha}) = \sum_{i,j=1}^{l} \alpha_i \, \alpha_j \, \mathcal{K}_{\sigma_x}(x_i, x_j) \, \mathcal{K}_{\sigma_y}(y_i, y_j) \tag{6}$$

is a norm in the reproducing kernel hilbert space and has an intuitive interpretation. Imagine the density is estimated by minimizing (6) only. Additionally, assume equidistant samples to be given. Then, (6) will penalize the variance in the weights of the samples. If the samples are not equidistant, the penalty will depend on the similarity of the samples. In order to avoid extreme outliers, the weights are bounded from above by a user-defined or automatically determined value ν.

The error term consists of slack variables measuring the error between the empirical cumulative distribution $\tilde{F}(x, y)$ and the respective cumulative distribution $F(x, y)$ of the joint density $f(x, y)$ over $\boldsymbol{x}$ and $\boldsymbol{y}$ based on the conditional density estimate, i.e.,

$$F(x, y) = \sum_{i=1}^{l} \frac{1}{l} \sum_{j=1}^{l} \mathcal{K}_{\sigma_x}(x_i, x_j) \, \mathrm{s}(x - x_j) \cdot \int_{-\infty}^{y} \mathcal{K}_{\sigma_y}(y_i, y') \, \mathrm{d}y' \, ,$$

with $\mathrm{s}(x) = 1$ if $x \geq 0$ and $\mathrm{s}(x) = 0$ otherwise. Note, that the error is compared at the sample points $\mathcal{D}$ only. It is assumed that $f(x) = \sum_{i=1}^{l} w_i \, \delta(x - x_i)$ is well-defined in the sense, that the data is spread evenly over the considered interval of $\boldsymbol{x}$. Here, $\delta(.)$ denotes the Dirac distribution. To allow for small levels of noise, the error terms $\xi_i^{(*)} := \{\xi_i\,, \xi_i^*\}$ measure the error exceeding an ε-insensitive loss zone only.

For estimating a function f based on $\mathcal{D}$ that is a valid conditional density function, the following constraints need to be asserted

$$\int_{-\infty}^{\infty} f(y|\hat{x})\,\mathrm{d}y = 1\,, \qquad f(y|\hat{x}) \geq 0\,, \tag{7}$$

for all fixed $\hat{x}$. For a mixture density of this kind, meeting the second constraint in (7) is simple, if the mixture components are Gaussian densities, by asserting all weights to be non-negative, i.e., $\alpha_i \geq 0$. Complying with the first constraint in (7) is hard, if not impossible, and will in general require an *ex-post* normalization step. In [7], the constraint

$$\sum_{i=1}^{l} \frac{1}{l} \sum_{j=1}^{l} \mathcal{K}_{\sigma_x}(x_j, x_i) = 1$$

is devised to approximate this constraint in (7). The rationale behind this approximation is to normalize the conditional density by normalizing the joint density of $f(x, y)$. As the only available information about the true density $f(x)$ is the empirical cumulative distribution $\tilde{F}(x)$, the constraint simplifies to a normalization for each conditional density $f(y|x_i)$ conditioned on each element of the training set, $i := 1, \ldots, l$. An alternative approximation can be sought if the samples are restricted to an interval $x_i \in [x_{\min}, x_{\max}]$ with $\mathrm{I} := x_{\max} - x_{\min}$. In this case, the normalization can be performed without the approximation by the empirical distribution $\tilde{F}(x)$, but one can apply the constraint

$$\int_{x_{\min}}^{x_{\max}} \int_{-\infty}^{\infty} f(y|x)\,\mathrm{d}y\,\mathrm{d}x = \mathrm{I}\,.$$

Empirically, it was observed that the following constraints lead to slightly better results

$$\sum_{i=1}^{l} \alpha_i = \mathrm{I}'\,, \qquad 0 \leq \alpha_i \leq \underbrace{\min\left(\mathrm{I}', \max(\,\nu, \mathrm{I}'/l\,)\right)}_{=:\nu'}\,. \tag{8}$$

with $i = 1, \ldots, l$ and $\mathrm{I}' := \frac{\mathrm{I}}{2\pi\,\sigma_x\,\sigma_y}$.

3.2 Optimization Problem

In order to arrive at a readily solvable optimization problem, the complexity penalty term, the error term, and the constraints (8) are combined. The trade-off between the error terms $\xi_i^{(*)}$ and the complexity penalty $\Omega(\underline{\alpha})$ is adjusted by a scalar parameter λ. This value is set before the optimization starts and reflects the user's confidence in how well the data represent the true underlying conditional density.

Estimating $f(y|x)$ corresponds to solving the following quadratic optimization problem given in the standard formulation

$$\begin{aligned} \min_{\alpha_{1:l},\, \xi_{1:l},\, \xi^*_{1:l}} \quad & \Omega(\underline{\alpha}) + \lambda \cdot \frac{1}{l} \sum_{i=1}^{l} (\xi_i + \xi_i^*) \\ \text{s.t.} \quad & \tilde{F}(x_i, y_i) - F(x_i, y_i) \leq \varepsilon + \xi_i\,, \\ & F(x_i, y_i) - \tilde{F}(x_i, y_i) \leq \varepsilon + \xi_i^*\,, \\ & \sum_{i=1}^{l} \alpha_i = \mathrm{I}'\,,\; 0 \leq \alpha_i \leq \nu'\,. \end{aligned} \tag{9}$$

The problem (9) may be solved using a standard quadratic program solver. The solution is a mixture of products of kernels centered at salient $(x_i, y_i) \in \mathcal{D}$—the support vectors (SVs). This optimization approach is a novel blend of Vapnik's approach to conditional density estimation [7] and a *Support Vector Regression* [8] with ε-insensitive loss function.

Note, that the algorithm above is restricted to scalar in- and output dimensions only. Extending the conditional density estimation to the multi-dimensional case is trivial, if the kernels for each dimension are separable. If this is not the case, artifacts arising from the ambiguity of the multi-variate empirical cumulative distribution function may make an application of a localized cumulative distribution [11] necessary. This will lead to major changes in the formulation of (9).

3.3 Hyper-Parameter Determination

Solving the above quadratic program requires setting the hyper-parameters

$$\theta := \{\sigma_x, \sigma_y, \nu, \lambda, \varepsilon\}\,.$$

To this end, an optimization scheme based on cross-validation is proposed. For each partitioning $\mathcal{D} = \mathcal{T} \cup \mathcal{V}$ of the data set, $\mathcal{T} \cap \mathcal{V} = \emptyset$ is assumed. Given an initial parameter estimate θ_0, the optimization problem (9) is solved for the training data $\mathcal{T}$. The resulting conditional density $f(.|.)$ is tested on the hold-out set $\mathcal{V}$. As a validation function $G(.)$, a weighted sum of the negative log-likelihood NL_x and a weight penalty

$$G(\mathcal{V}) = \|\ \underline{\alpha}\ \|_2^2 + C \cdot \sum_{v \in \mathcal{V}} \log f(v|\theta_l) \tag{10}$$

is employed. For k-fold cross-validation, the results are averaged to assess the quality. The hyper-parameters may then be found by minimizing the average $G(\mathcal{D})$ by standard methods, e.g., a (Quasi-)Newton approach. Note (10) is subject to constraints, e.g., $\sigma > 0$ and valid ν. Domain knowledge may be introduced by constraining the parameters and determining the trade-off C between data fit and variance in the weights, which resembles $\Omega(\underline{\alpha})$ but is cheaper to compute. For example, in the experiments in Sec. 5, tight bounds on ε and λ were user-defined or obtained from a line-search. Alg. 1 summarizes the results of this section.

Algorithm 1 Conditional Density Estimation

```
Input: $\mathcal{D}$, $\theta_0$
  $\theta_l := \theta_0$
  while Gradient $>$ Threshold do
    for all Partitions $i$ do
      Solve optimization problem (9) for $\mathcal{T}_i$, $\theta_l \rightarrow f$
      Validate $f$ using $\mathcal{V}_i$, $\theta_l \rightarrow G_i$
    end for
    Average all $G_i \rightarrow G$
    Update $\theta_l$: $\underset{\theta_l}{\text{minimize}}\ G \rightarrow \theta_{l+1}$
  end while
```

4 Support-Vector Density Filter (SVDF)

In this section, it will be shown how the derived SV conditional densities can be used with the prediction and filter step of Sec. 2. The actual steps resemble the processing for mixtures of axis-aligned normal densities [9, 10], but require additional normalization. The resulting filter is a good compromise with regard to the general issues of nonlinear filtering addressed in Sec. 2.

4.1 Prediction Step

A prior distribution is assumed to be given by

$$f(x_{k-1}|\hat{y}_{1:k-1}) = \sum_{e\in\mathcal{E}} \alpha_e\, \mathcal{K}_{\sigma_e}(x_{k-1}, x_{k-1}^e)\,. \tag{11}$$

For the sake of brevity, the indexing of weights, kernels, and parameters is summarized by index e, which corresponds to a specific mixture component. Furthermore, let the result of estimating $f(x_k|x_{k-1})$ on the basis of a data set $\mathcal{D}$, by solving (9), be given in the form of

$$f(x_k|x_{k-1}) = \sum_{a\in\mathcal{A}} \overline{\alpha}_a\, \mathcal{K}_{\sigma_a^{(1)}}(x_{k-1}, x_{k-1}^a)\, \mathcal{K}_{\sigma_a^{(2)}}(x_k, x_k^a)\,. \tag{12}$$

In the above equation, the means x_{k-1}^a and x_k^a correspond to support vectors from the training set $\mathcal{D}$. After inserting $f(x_{k-1}|\hat{y}_{1:k-1})$ in the form of (11) and (12), one may simplify (3) after some rearrangements to

$$f(x_k|\hat{y}_{1:k-1}) = \sum_{a\in\mathcal{A}} \overline{\alpha}_a\, \mathcal{K}_{\sigma_a^{(2)}}(x_k, x_k^a)$$
$$\cdot \sum_{e\in\mathcal{E}} \alpha_e \underbrace{\int \mathcal{K}_{\sigma_a^{(1)}}(x_{k-1}, x_{k-1}^a)\, \mathcal{K}_{\sigma_e}(x_{k-1}, x_{k-1}^e)\, \mathrm{d}x_{k-1}}_{=:\, c^{ap}\, \mathcal{K}_{\sigma_{ap}}(x_{k-1}^a, x_{k-1}^e)}\,.$$

Defining $\sigma_{ap} := \sqrt{\left(\sigma_a^{(1)}\right)^2 + \sigma_e^2}$, $x_k^p := x_k^a$, $\mathcal{P} := \mathcal{A}$, $\sigma_p := \sigma_a^{(2)}$ and setting the new weights to

$$\alpha_p := \overline{\alpha}_a\ c^{ap} \sum_{e\in\mathcal{E}} \alpha_e\, \mathcal{K}_{\sigma_{ap}}(x_{k-1}^a, x_{k-1}^e)\,,$$

gives the distribution of the predicted state $\boldsymbol{x}_k$. The resulting density

$$f(x_k|\hat{y}_{1:k-1}) = \sum_{p\in\mathcal{P}} \alpha_p\, \mathcal{K}_{\sigma_p}(x_k, x_k^p) \tag{13}$$

can be understood as the result of weighting the mixture density about $\boldsymbol{x}_k$ with the product of the prior density and the x_{k-1}-dimension of the transition density. It is noteworthy that $|\mathcal{P}|$ in (13) is constant and depends on the number of support vectors only. Thus, every prediction step will return a mixture with a fixed number of components. The calculations necessary to obtain (13) can be performed analytically and exactly.

4.2 Filter Step

In order to solve (4), we assume $f(x_k|\hat{y}_{1:k-1})$ to be given in the form of (13) and the likelihood function $f(y_k|x_k)$ to be given in form of a solution to (9), i.e.,

$$f(y_k|x_k) = \sum_{h\in\mathcal{H}} \overline{\alpha}_h\, \mathcal{K}_{\sigma_h^{(1)}}(x_k, x_k^h)\, \mathcal{K}_{\sigma_h^{(2)}}(y_k, y_k^h)\,. \tag{14}$$

In the above equation the means y_k^h and x_k^h correspond to support vectors from the training set $\mathcal{D}$. Inserting (13), (14) and a measurement value $\hat{y}_k$ into (4) yields

$$f(x_k|\hat{y}_{1:k}) = c \cdot \left[\sum_{h\in\mathcal{H}} \overline{\alpha}_h\, \mathcal{K}_{\sigma_h^{(1)}}(x_k, x_k^h)\, \mathcal{K}_{\sigma_h^{(2)}}(\hat{y}_k, y_k^h)\right]$$
$$\cdot \left[\sum_{p\in\mathcal{P}} \alpha_p\, \mathcal{K}_{\sigma_p}(x_k, x_k^p)\right]\,. \tag{15}$$

Combining the sums in (15) and rearranging gives

$$f(x_k|\hat{y}_{1:k}) = c \sum_{e\in\mathcal{E}} \alpha_e\, \mathcal{K}_{\sigma_e}(x_k, x_k^e)\,. \tag{16}$$

For $e := (h, p)$, i.e., $\mathcal{E} = \mathcal{P} \times \mathcal{H}$ with $\sigma_e := \sqrt{\frac{\left(\sigma_h^{(1)}\right)^2 \sigma_p^2}{\left(\sigma_h^{(1)}\right)^2 + \sigma_p^2}}$, one obtains

$$\alpha_e := \overline{\alpha}_h\, \alpha_p\, \mathcal{K}_{\sigma_h^{(2)}}(\hat{y}_k, y_k^h)\ \mathcal{K}_{\sigma_e}(x_k^p, x_k^h)\,,$$
$$x_k^e := \frac{x_k^p \left(\sigma_h^{(1)}\right)^2 + x_k^h \sigma_p^2}{\left(\sigma_h^{(1)}\right)^2 + \sigma_p^2}\,.$$

The normalization constant is $c := f(\hat{y}_k|\hat{y}_{1:k-1}) = \frac{1}{\sum_{e\in\mathcal{E}} \alpha_e}$. The expression in (16) can be calculated analytically and exactly. Since $|\mathcal{E}| = |\mathcal{P}| \cdot |\mathcal{H}|$, repeated filter steps will cause an exponential increase in the number of components. Yet, if $f(x_k|\hat{y}_{1:k})$ is processed in the prediction step, the number of components $|\mathcal{E}|$ will remain constant. The proposed SVDF allows for exact analytic computation of the prediction and filter step. One obtains posterior distributions in the form of mixtures of axis-aligned Gaussian kernels allowing for multimodal state distributions. An extension to the multi-dimensional case is trivial, if the in- and output dimensions are assumed to be separable. In case of repeated filtering without intermediate prediction, i.e., fusing several measurements that are mutually conditionally independent given the state $\boldsymbol{x}_k$, an exponential growth in the number of components will occur. If the computational burden is too high, standard fast Gaussian mixture reduction algorithms, e.g., [12, 13, 14], may be employed.

5 Experiments

In order to assess the quality of the proposed approach as a stand-alone conditional density estimation procedure, likelihood scores were compared with results of Expectation Maximization [15] for Gaussian mixture densities, kernel density estimation (KDE), and Gaussian Process Regression (GPR) [3]. EM and KDE can be understood as the default density estimators, whereas GPR has been shown to produce good probabilistic models for filtering applications with underlying functional dependency.

In order to evaluate the performance on a problem with nonlinear system and measurement models, the SVDF is compared with EKF, UKF, GP-UKF, and GP-ADF on the growth process example from [6]. The above filters assume the density to be well-represented by a Gaussian. In contrast, the SVDF is capable of maintaining multimodal posteriors, which will be shown by the application to the cubic system function problem, as introduced in Sec. 5.1. Additionally, results for the case of a linear system and a nonlinear measurement model are presented, which typically produces singularities in GP-based Filters such as the GP-ADF. For the experiments, the Matlab$^{\text{TM}}$ implementation of EM, the kernel density estimation (KDE) toolbox [16] and the EKF, UKF, GP-UKF, as well as GP-ADF implementations from [6] were used.

5.1 Comparison of Likelihood Scores

For comparing the quality of the conditional density estimation produced by the proposed approach, conditional densities were generated based on samples drawn from a cubic function disturbed by additive noise

$$\boldsymbol{x}_{k+1} = 2\,\boldsymbol{x}_k - 0.5\,\boldsymbol{x}_k^3 + \boldsymbol{w}_k\,, \;\; \boldsymbol{w}_k \sim \mathcal{N}(0, 0.175)\,. \quad (17)$$

For training, 100 points were randomly distributed in $[-3, 3]$ and another 100 points were generated as a hold-out set. The quality is measured by the likelihood of this hold-out set, where a uniform prior is assumed. Tab. 1 lists the negative log-likelihood NL_x of the test data. Lower values indicate higher estimation quality. Two configurations of EM were used. As we assume additive noise, EM only trains mixture models of Gaussians with covariance matrices that are non-zero only on the main diagonal. In the first configuration, EM was restricted to estimating homoscedastic Gaussian mixture densities, i.e., a mixture with identical variances σ_x and σ_y, only. The number of Gaussians to be fit was set to the number of Gaussians minus one[1] obtained by the SVR. In the last EM-experiment, again homoscedastic Gaussian mixture densities (EM2) were estimated, but the number of Gaussians was obtained trying all numbers smaller than the number of samples and choosing the model with the best Akaike information criterion score. Regarding non-parametric approaches, results for a kernel density estimator (KDE) with bandwiths chosen according to *rule-of-thumb* [17], a reduced kernel density estimate (RSDE) [18], and GPR are reported. The conditional densities for EM, KDE, and RSDE are obtained on the basis of the obtained joint density estimates.

	EM1	EM2	KDE	RSDE	GPR	SVR
NL_x	1.90	2.79	0.87	1.03	0.14	0.55
Comp	87.2	14.6	100	11.1	100	88.2

Table 1: Negative log-likelihood of the test data and number of the components for the conditional density estimates returned by EM, GPR and the proposed quadratic programming approach. The results are averages over ten experiments.

The results in Tab. 1 show that EM produces the worst conditional densities. The reason for this is shown in Fig. 1. The conditional densities returned by EM suffer from overfitting in the extreme points of (17) around $(\pm\sqrt{2}, \pm\sqrt{2})$. All non-parametric approaches produce conditional densities that appear to have an underlying continuous function, whereas GPR and SVR clearly outperform KDE and RSDE. Note that the GPR results are misleadingly good as the system chosen is especially favorable for this technique. The GPR's problems with other system types and the advantages of the SVR's mixture density representation for filtering applications are discussed in the following experiments.

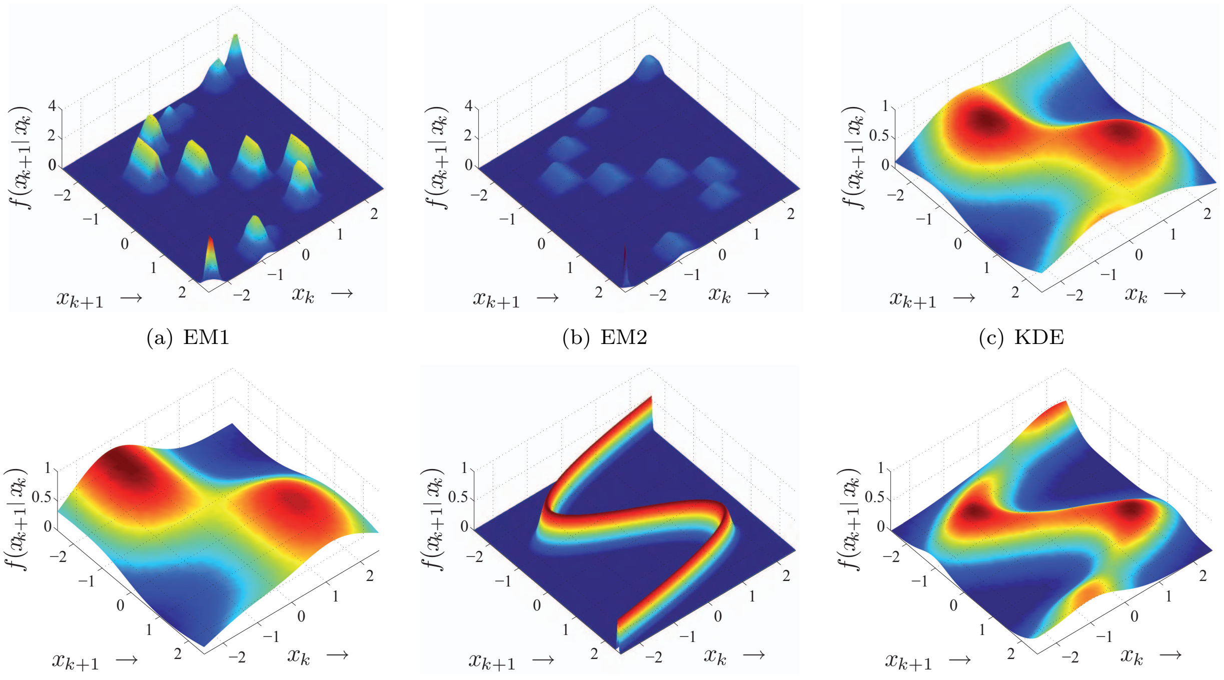
5.2 Growth Process

Given the scalar nonlinear system and measurement equations

$$\begin{aligned} \boldsymbol{x}_{k+1} &= 0.5\,\boldsymbol{x}_k + \tfrac{25\,\boldsymbol{x}_k}{1+\boldsymbol{x}_k^2} + \boldsymbol{w}_k\,, & \boldsymbol{w}_k &\sim \mathcal{N}(w_k, 0.2)\,, \\ \boldsymbol{y}_{k+1} &= 5\,\sin(2\,\boldsymbol{x}_{k+1}) + \boldsymbol{v}_{k+1}\,, & \boldsymbol{v}_{k+1} &\sim \mathcal{N}(v_{k+1}, 0.01)\,, \end{aligned}$$

the hidden state shall be estimated. As a kernel, $\mathcal{N}(x - \hat{x}, \sigma) := \frac{1}{\sqrt{2\pi}\sigma} \cdot \mathcal{K}_\sigma(x, \hat{x})$ is used. The problem resembles a growth model [19] and is taken from [6]. For training, 100 points were randomly distributed in $[-10, 10]$. The prior density is given as a normal density with $\mu_0 \in [-10, 10]$ and $\sigma_0 = 0.5$. For 200 independent states $x_0^{(i)}$, the observations $y_1^{(i)}$ of the successive states were calculated, where $i = 1, \ldots, 100$. The

[1] This is due to numerical issues with the EM implementation.



(a) EM1

(b) EM2

(c) KDE

(d) RSDE

(e) GPR

(f) SVR

Figure 1: From left to right: EM results with different number of components and dense as well as sparse kernel density estimates. The last two results are obtained from GPR and SVR. Note that the densities were normalized except for the GPR results, which are automatically normalized.

performance is assessed by comparing the Mahalanobis distance $\mathcal{M}(x)$ of the filtered mean to the ground truth and negative log-likelihood NL_x of the hidden state in ten experiments. For the Gaussian mixture densities (SVDF, SVDF*) $\mathcal{M}(x)$ and NL_x were calculated for a moment-matched Gaussian density. Even though lower values of $\mathcal{M}(x)$ and NL_x indicate better accuracy, only NL_x penalizes uncertainty. To exemplify the differences, Fig. 2 gives the results for the above generative models for one run, when setting the seed for the random samples identical to [6]. The averaged results over ten experiments are given in Fig. 3(a) and show that the estimation performance of the SVDF compares well to the GP-ADF's performance.

Note that both probabilistic models used with the SVDF were determined by hyper-parameter optimization. The results for SVDF* are obtained by manually tuning the hyper-parameters. For the example in Fig. 2, the number of SVs returned by the SVR for automatically and manually obtained hyper-parameters are given in the following table. Apparently, the conditional density for the system model is harder to estimate than for the measurement model.

% SVs (100)	Automatic	Manual
$f(x_{k+1}\|x_k)$	100	100
$f(y_k\|x_k)$	31	56

5.3 Cubic System Function

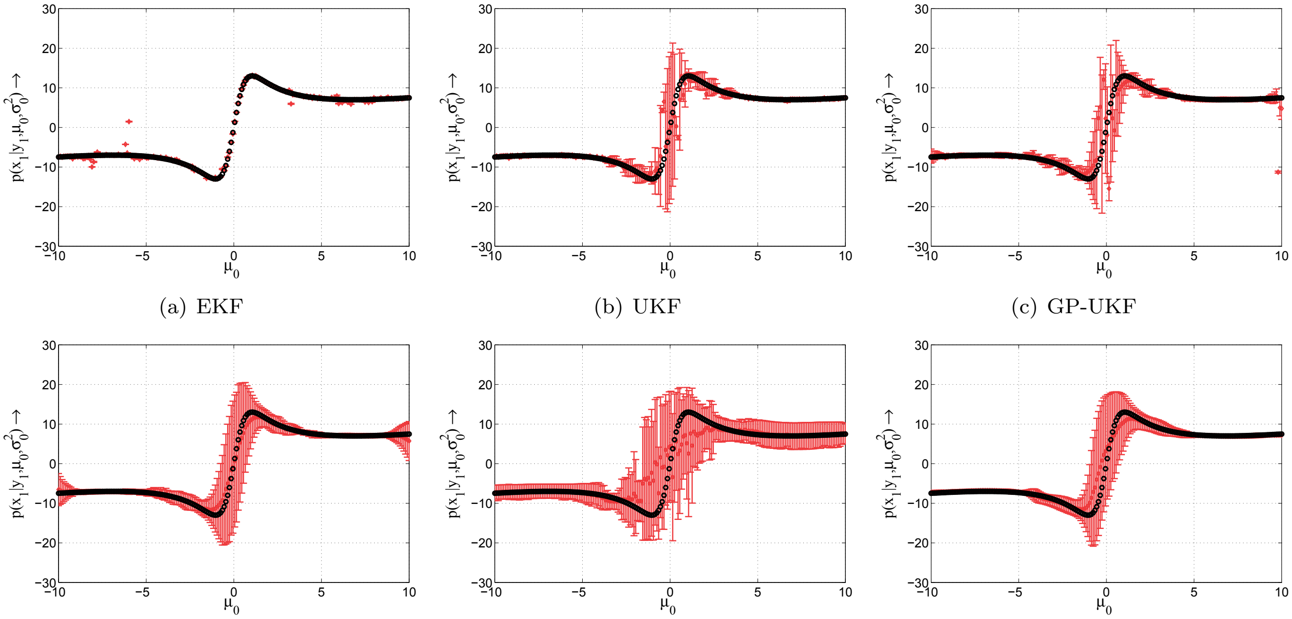
In order to assess the filter's capability of maintaining multimodal posterior densities, the results of successive predictions using the cubic system function (17) are reported. The prior density is given by $f(x_0) = \mathcal{N}(x_0 - 0.4, 0.8)$. Fig. 4 shows the true posterior density as well as the GP-ADF and SVDF estimated posterior densities after 1, 2, 3, and 4 prediction steps using (17) and the above prior. Furthermore, a moment-matched Gaussian approximation to the SVDF posterior is depicted. The true posterior was obtained from high resolution numerical integration. These results explicate the SVDF's possibility of estimating multimodal posterior densities. The EKF, UKF, GP-UKF, and GP-ADF only support Gaussian posterior densities and thus, are not suitable for this type of problem. Even the moment-matched Gaussian approximation to the SVDF's posterior density yields better results.

5.4 Recursive Filtering

Recursive filtering of a time series governed by

$$\begin{aligned} \boldsymbol{x}_{k+1} &= 1.1\,\boldsymbol{x}_k + w\,, & \boldsymbol{w}_k &\sim \mathcal{N}(w_k, 0.25)\,, \\ \boldsymbol{y}_{k+1} &= \left(\tfrac{\boldsymbol{x}_{k+1}}{10}\right)^3 + v\,, & \boldsymbol{v}_{k+1} &\sim \mathcal{N}(v_{k+1}, 0.25)\,. \end{aligned}$$

is considered. This corresponds to a linear system and a nonlinear measurement model. The prior density is $f(x_0) = \mathcal{N}(x_0 - 1, 2)$. Fig. 3(b) gives the true and estimated state trajectory for GP-ADF and SVDF. The GP-based filter encounters a singularity, which is not



(a) EKF

(b) UKF

(c) GP-UKF

(d) GP-ADF

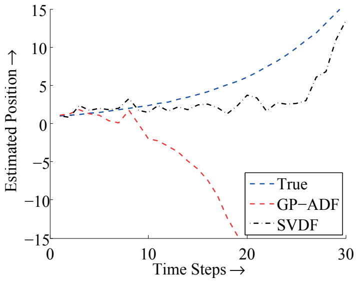
(e) SVDF

(f) SVDF*

Figure 2: True hidden states (black) as well as mean and 2σ bounds (red) for the EKF, UKF, GP-ADF, GP-UKF, the automatically and a manually tuned SVDF. The results $f(x_1|y_1,\mu_0,\sigma_0^2)$ are given for varying mean values μ_0 of $f(x_0)$. Note that for the above results the identical seed for the random samples as in [6] was used.

	$NL_x^{0.25}$	$NL_x^{0.50}$	$NL_x^{0.75}$	$\mathcal{M}(x)$
EKF	888.53	29472.41	276614.34	2073897.77 ± 2964656.78
UKF	61.35	605.7606	2383.86	1042.39 ± 4588.72
GP-UKF	62.68	424.5677	1692.86	1790.40 ± 16548.19
GP-ADF	59.31	276.8323	1050.36	21.19 ± 32.42
SVDF	59.96	174.7015	379.27	1.30 ± 1.00
SVDF*	64.27	366.0246	1064.31	20.31 ± 21.20

(a) Average negative log-likelihood NL_x of the hidden state and Mahalanobis distance $\mathcal{M}(x)$ of the filtered mean compared to the ground truth for the EKF, UKF, GP-ADF, GP-UKF, and SVDF with automatically and manually obtained hyper-parameters.



(b) Estimated means for GP-ADF and SVDF as well as the true state for 30 time steps, cf. Sec. 5.4.

Figure 3: Results for the experiments in Sec. 5.2 and Sec. 5.4, respectively.

seldom when linear system models are used. As can be seen, the SVDF tracks the true system correctly and additionally does not suffer from singularities. The SVDF's system model contains 56 of 100 samples and the measurement model contains 31 of 100 samples.

6 Conclusion

In this paper, a conditional density estimation algorithm for nonlinear stochastic dynamic systems was proposed. A novel support vector regression was introduced by which conditional densities representing system and measurement function can be obtained based on samples only. The regression was stated in form of a novel quadratic program based on an ε-insensitive loss function, practical normalization constraints, and anisotropic variances. For automatically determining the hyper-parameters, an algorithm based on cross-validation was devised. Employing the sample-based densities, an existing axis-aligned Gaussian mixture filter was modified and analytic expressions for the prediction and filtering step were derived. Benchmark experiments show the filter to provide good accuracy. The filter supports multimodal densities and does not suffer from singularities in case of linear system models as recent *Gaussian Process*-based filter algorithms do.

So far, the proposed approach is restricted to scalar in- and output dimensions. It remains future work to extend the presented work to the vector-valued case. Furthermore, it is imaginable that even sparser conditional densities can be obtained for improving the filter's runtime performance even more. The hyper-parameter determination algorithm might be improved

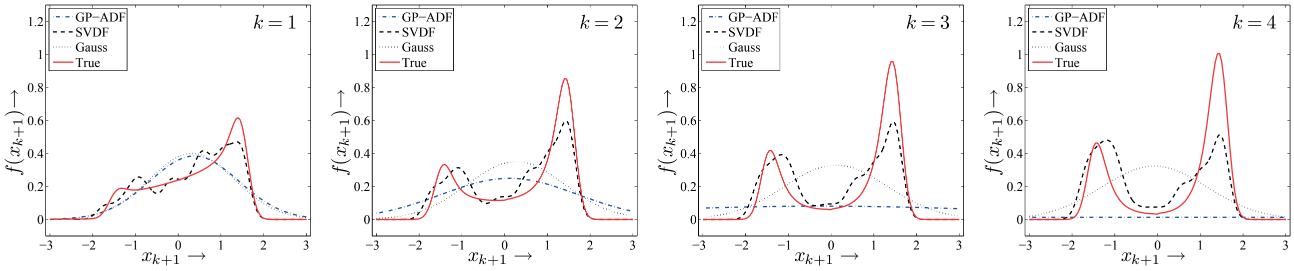


Figure 4: Left to right: True (red, solid) posterior densities, the GP-ADF estimated posterior densities (blue, dash-dotted), the SVDF estimated posterior densities (black, dash), and moment-matched Gaussian approximation of the SVDF estimate (black, dotted) after 1,2,3, and 4 prediction steps.

w.r.t. efficiency and effectiveness. Regarding online adaptation the use of incremental *Expectation Maximization* or SVR with uncertain data, i.e., no certain label, should be investigated. Additionally, resolving the restriction to kernels centered about the support vectors seems promising.

References

[1] D. Simon, *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*, Wiley & Sons, first edition, 2006.

[2] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[3] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, Massachusetts, 2006.

[4] J. Ko, D. Klein, D. Fox, and D. Haehnel, "GP-UKF: Unscented Kalman Filters with Gaussian Process Prediction and Observation Models," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, California, October 2007, pp. 1901–1907.

[5] J. Ko and D. Fox, "GP-BayesFilters: Bayesian Filtering Using Gaussian Process Prediction and Observation Models," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, September 2008, pp. 3471–3476.

[6] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic Moment-based Gaussian Process Filtering," in *26th International Conference on Machine Learning (ICML)*, Montreal, Canada, June 2009.

[7] V. Vapnik, *The Nature of Statistical Learning Theory*, Statistics for Engineering and Information Science. Springer, New York, 2. edition, 2000.

[8] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett, "New Support Vector Algorithms," *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[9] E. Driver and D. Morrell, "Implementation of Continuous Bayesian Networks Using Sums of Weighted Gaussians," in *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Montreal, Canada, August 1995, pp. 134–140.

[10] M. Huber, D. Brunn, and U. D. Hanebeck, "Closed-Form Prediction of Nonlinear Dynamic Systems by Means of Gaussian Mixture Approximation of the Transition Density," in *Proceedings of the 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Heidelberg, Germany, September 2006, pp. 98–103.

[11] U. D. Hanebeck and V. Klumpp, "Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance," in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Republic of Korea, August 2008, pp. 33–39.

[12] M. F. Huber and U. D. Hanebeck, "Progressive Gaussian Mixture Reduction," in *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, Cologne, Germany, July 2008, pp. 1–8.

[13] A. R. Runnalls, "Kullback-Leibler Approach to Gaussian Mixture Reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, July 2007.

[14] M. West, "Approximating Posterior Distributions by Mixtures," *Journal of the Royal Statistical Society: Series B*, vol. 55, no. 2, pp. 409–422, 1993.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the *EM* Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[16] A. Ihler, "Kernel Density Estimation Toolbox for Matlab," 2003.

[17] D. W. Scott, *Multivariate Density Estimation : Theory, Practice, and Visualization*, Wiley Series in Probability and Mathematical Statistics - A Wiley Interscience publication. Wiley, New York, 1992.

[18] M. Girolami and C. He, "Probability Density Estimation from Optimally Condensed Data Samples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1253–1264, 2003.

[19] G. Kitagawa, "Monte Carlo Filter and Smoother for non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.