Overlapping Mixtures of Gaussian Processes for the Data Association Problem

Miguel Lázaro-Gredilla[a,*], Steven Van Vaerenbergh[a], Neil Lawrence[b]

[a] *Dept. Communications Engineering, University of Cantabria, 39005 Santander, Spain*
[b] *Dept. of Computer Science, University of Sheffield, S1 4DP Sheffield, UK*

Abstract

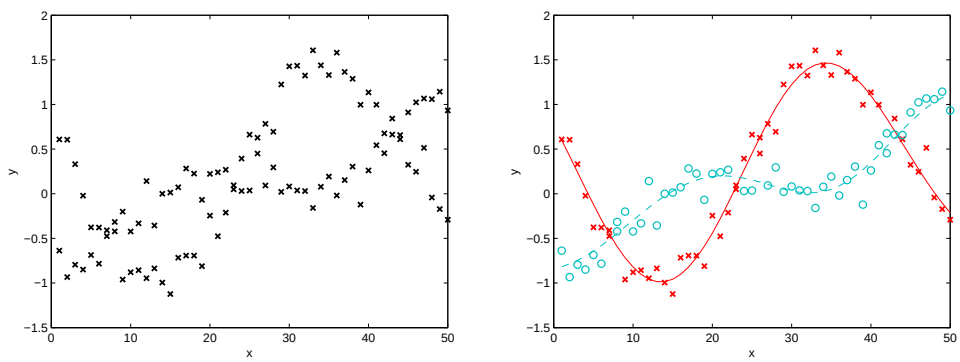
In this work we introduce a mixture of GPs to address the data association problem, i.e. to label a group of observations according to the sources that generated them. Unlike several previously proposed GP mixtures, the novel mixture has the distinct characteristic of using no gating function to determine the association of samples and mixture components. Instead, all the GPs in the mixture are global and samples are clustered following "trajectories" across input space. We use a non-standard variational Bayesian algorithm to efficiently recover sample labels and learn the hyperparameters. We show how multi-object tracking problems can be disambiguated and also explore the characteristics of the model in traditional regression settings.

Keywords: Gaussian Processes, Marginalized Variational Inference, Bayesian Models

*Corresponding author: Tel: +34 942200919 ext 802, Fax: +34 942201488.
Email addresses: `miguellg@gtas.dicom.unican.es` (Miguel Lázaro-Gredilla), `steven@gtas.dicom.unican.es` (Steven Van Vaerenbergh), `N.Lawrence@sheffield.ac.uk` (Neil Lawrence)

1. Introduction

The data association problem arises in multi-target tracking scenarios. Given a set of observations that represent the positions of a number of moving sources, such as cars or airplanes, data association consists of inferring which observations originate from the same source [1, 2]. Data association is found in tracking problems for instance in computer vision [3], surveillance, sensor networks [4] and radar tracking [5]. An example of data association with two sources is illustrated in Figure 1.



(a) One-dimensional observations.

(b) Solution obtained by the proposed method.

Figure 1: Example of a multi-target tracking scenario. Data association aims to identify what observations correspond to each source.

For a human observer, little effort is required to distinguish two noisy trajectories in this example, representing the paths followed by two objects in time. In this specific case, one observation of each target is available at each time instant, and the measurement instants are equally spaced in time, although neither of these properties are required in general.

Typical multi-target tracking algorithms operate online. They include joint Kalman filters [6] and joint particle filters [7]. Given the predicted positions of the targets and a number of candidate observed positions, they usually make instant data association decisions based on nearest-neighbor criteria or statistically more sophisticated approaches such as the Joint Probabilistic Data-Association Filter (JPDAF) [5, 7] or the Multiple Hypothesis Tracker (MHT) [6]. An important disadvantage of these classical techniques is that they usually require to determine a large number of parameters. This drawback motivated the development of several conceptually simpler approaches based on motion geometry heuristics [2, 8, 9]. However, these approaches are usually limited to specific scenarios, and they show difficulties in the presence of noise and when several trajectories cross each other.

Most data association techniques can be significantly improved by postponing decisions until enough information is available to exclude ambiguities [2], although this causes the number of possible trajectories to grow exponentially. Some attempts have been made to restrain this combinatorial explosion, including the heuristic methods from [10, 11].

In this paper we present an algorithm based on Gaussian Processes that is able to consider all available data points in batch form whilst avoiding the exponential growth in potential tracks. As a result, it is capable to deal with difficult data association problems in which trajectories come very close and even cross each other. Furthermore, the algorithm does not require any knowledge about the model underlying the data, and it does not need time instants to be evenly spaced, nor to contain observations from all sources.

Gaussian Processes (GPs) [12] are a powerful tool for Bayesian nonlin-

ear regression. When combined in mixture models, GPs can be applied to describe data where there are local non-stationarities or discontinuities [13, 14, 15, 16]. The components of the mixture model are GPs and the prior probability of any given component is typically provided by a gating function. The role of the gating function is to dictate which GP is a priori most likely to be responsible for the data in any given region of the input space, i.e., the gating network forces each component of the GP mixture to be localized.

In this work we follow a different approach, inspired by the data association problem. In particular, for any given location in input space there may be multiple targets, perhaps corresponding to multiple objects in a tracking system. We are interested in constructing a GP mixture model that can associate each of these targets with separate components. When there is ambiguity, the posterior distribution of targets will reflect this. We therefore propose a simple mixture model in which each component is global in its scope. The assignment of the data to each GP is performed sample-wise, independently of input space localization. In other words, no gating function is used. We call this model the Overlapping Mixture of GPs (OMGP).

It has been brought to our attention that the proposed model bears resemblance with the work of [17]. However, the focus of application is clearly different. In [17], the objective is to cluster a set of trajectories according to their similarity, whereas in this work we tackle the task of clustering observations into trajectories (a more demanding task, since only single observations, as opposed to full trajectories, are available). Also, [17] uses a standard variational Bayesian algorithm, whereas in this work we take ad-

vantage of non-standard variational algorithms [18, 19] to derive a tighter bound.

The remainder of this paper is organized as follows: In Section 2 we provide a brief review of GPs in the regression setting. Section 3 first introduces the OMGP model and then discusses how to perform efficient learning, hyperparameter selection, and predictions using this model. Experiments on several data sets are provided in Section 4. We wrap up in Section 5 with a brief discussion.

2. Brief Review of Gaussian Processes

In recent years, Gaussian Processes (GPs) have attracted a lot of attention due to their nice analytical properties and their state-of-the-art performance in regression tasks (see [20]). In this section we provide a brief summary of the main results for GP regression, see [12] for further details.

Assume that a set of N multi-dimensional inputs and their corresponding scalar outputs, $\mathcal{D} \equiv \{\mathbf{x}_n, y_n\}_{i=1}^m$, are available. The regression task is, given a new input $\mathbf{x}_*$, to obtain the predictive distribution for the corresponding observation y_* based on $\mathcal{D}$.

The GP regression model assumes that the observations can be modeled as some noiseless latent function of the inputs plus independent noise $y = f(\mathbf{x}) + \varepsilon$, and then sets a zero-mean[1] GP prior on the latent function $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ and a Gaussian prior on $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ on the noise, where

[1] To make this assumption hold, the sample mean of the set $\{y(\mathbf{x}_n)\}_{n=1}^m$ is usually subtracted from data before proceeding further.

$k(\mathbf{x}, \mathbf{x}')$ is a covariance function and σ^2 is a hyperparameter that specifies the noise power.

The covariance function $k(\mathbf{x}, \mathbf{x}')$ specifies the degree of coupling between $y(\mathbf{x})$ and $y(\mathbf{x}')$, and it encodes the properties of the GP such as power level, smoothness, etc. One of the best-known covariance functions is the anisotropic squared exponential. It has the form of an unnormalized Gaussian, $k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 \exp\left(-\frac{1}{2}\mathbf{x}^\top \mathbf{\Lambda}^{-1}\mathbf{x}\right)$ and depends on the signal power σ_0^2 and the length-scales $\mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix containing one length-scale per input dimension. Each length-scale controls how fast the correlation between outputs decays as the separation along the corresponding input dimension grows. We will collectively refer to all kernel parameters as $\boldsymbol{\theta}$.

The joint distribution of the available observations (collected in $\mathbf{y}$) and some unknown output $y(\mathbf{x}_*)$ is a multivariate Gaussian distribution, with parameters specified by the covariance function:

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma^2 \mathbf{I}_N & \mathbf{k}_* \\ \mathbf{k}_*^\top & k_{**} + \sigma^2 \end{bmatrix}\right) , \tag{1}$$

where $[\mathbf{K}]_{nn'} = k(\mathbf{x}_n, \mathbf{x}_{n'})$, $[\mathbf{k}_*]_n = k(\mathbf{x}_n, \mathbf{x}_*)$ and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. $\mathbf{I}_N$ is used to denote the identity matrix of size N. The notation $[\mathbf{A}]_{nn'}$ refers to entry at row n, column n' of $\mathbf{A}$. Likewise, $[\mathbf{a}]_n$ is used to reference the n-th element of vector $\mathbf{a}$.

From (1) and conditioning on the observed training outputs we can obtain the predictive distribution

$$p_{\mathrm{GP}}(y_*|\mathbf{x}_*, \mathcal{D}) = \mathcal{N}(y_*|\mu_{\mathrm{GP}*}, \sigma^2_{\mathrm{GP}*}) \tag{2}$$

$$\mu_{\mathrm{GP}*} = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1}\mathbf{y} \qquad \sigma^2_{\mathrm{GP}*} = \sigma^2 + k_{**} - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1}\mathbf{k}_* ,$$

which is computable in $\mathcal{O}(N^3)$ time, due to the inversion[2] of the $N \times N$ matrix $\mathbf{K} + \sigma^2 \mathbf{I}_N$.

Hyperparameters $\{\boldsymbol{\theta}, \sigma\}$ are typically selected by maximizing the marginal likelihood (also called "evidence") of the observations, which is

$$\log p(\mathbf{y}|\boldsymbol{\theta}, \sigma) = -\frac{1}{2}\mathbf{y}^\top \left(\mathbf{K} + \sigma^2 \mathbf{I}_N\right)^{-1} \mathbf{y} - \frac{1}{2}|\mathbf{K} + \sigma^2 \mathbf{I}_N| - \frac{N}{2}\log(2\pi) \ . \quad (3)$$

If analytical derivatives of (3) are available, optimization can be carried out using gradient methods, with each gradient computation taking $\mathcal{O}(N^3)$ time. GP algorithms can typically handle a few thousand data points on a desktop PC.

When dealing with multi-output functions, instead of a single set of observations $\mathbf{y}$, D sets are available, $\mathbf{y}_1 \ldots \mathbf{y}_D$, each corresponding to a different output dimension. In this case we can assume independence across the outputs and perform the above procedure independently for each dimension. This will provide reasonable results for most problems, but if correlation between different dimensions is expected, we can take advantage of this knowledge and model them jointly using multi-task covariance functions [21].

3. Overlapping Mixtures of Gaussian Processes (OMGP)

The overlapping mixture of Gaussian processes (OMGP) model assumes that there exist M different latent functions $\{f^{(m)}(\mathbf{x})\}_{m=1}^{M}$ (which we will call "trajectories"), and that each output is produced by evaluating one of

[2] Of course, in a practical implementation, this inversion should never be performed explicitly, but through the use of the Cholesky factorization and the solution of the corresponding linear systems, see [12].

these functions at the corresponding input and by adding Gaussian noise to it. The association between samples and latent functions is determined by the $N \times M$ binary indicator matrix $\mathbf{Z}$: Entry $[\mathbf{Z}]_{nm}$ being non-zero specifies that n-th data point was generated using trajectory m. Only one non-zero entry per row is allowed in $\mathbf{Z}$.

To model multi-dimensional trajectories (i.e., when the mixture model has multiple outputs), D latent functions per trajectory can be used $\{f_d^{(m)}(\mathbf{x})\}_{m=1,d=1}^{M,D}$. Note that there is no need to extend $\mathbf{Z}$ to specifically handle the multi-output case, since all the outputs corresponding to a single input are the same data point and must belong to the same trajectory.

For convenience we will collect all the outputs in a single matrix $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_D]$ and all the latent functions of trajectory m in a single matrix $\mathbf{F}^{(m)} = [\mathbf{f}_1^{(m)} \dots \mathbf{f}_D^{(m)}]$. We will refer to all the latent functions as $\{\mathbf{F}^{(m)}\}$.

Given the above description, the likelihood of the OMGP model is

$$p(\mathbf{Y}|\{\mathbf{F}^{(m)}\}, \mathbf{Z}) = \prod_{n=1,m=1,d=1}^{N,M,D} \mathcal{N}([\mathbf{Y}]_{nd}|[\mathbf{F}^{(m)}]_{nd}, \sigma^2)^{[\mathbf{Z}]_{nm}} . \tag{4}$$

Following the standard Bayesian framework, we place priors on the unobserved latent variables

$$p(\mathbf{Z}) = \prod_{n=1,m=1}^{N,M} [\mathbf{\Pi}]_{nm}^{[\mathbf{Z}]_{nm}}, \qquad p(\mathbf{F}^{(m)}|\mathbf{X}) = \prod_{m=1,d=1}^{M,D} \mathcal{N}(\mathbf{f}_d^{(m)}|\mathbf{0}, \mathbf{K}^{(m)}) , \tag{5}$$

i.e., a multinomial distribution over the indicators (in which $\sum_{m=1}^{M} [\mathbf{\Pi}]_{nm} = 1 \quad \forall_n$) and independent GP priors over each latent function.[3] We allow

[3] If correlation between different trajectories is known to exist, trajectories can be jointly modeled as a single GP, using a covariance function that accounts for this dependence.

different covariance matrices for each trajectory. Though the multinomial distribution is specified here in its more general form, additional constraints are usually imposed, such as holding the prior probabilities constant for all data points. For the sake of clarity, we will omit the conditioning on the hyperparameters $\{\boldsymbol{\theta}, \boldsymbol{\Pi}, \sigma^2\}$, which can be assumed to be known for the moment.

Unfortunately, the analytical computation of the posterior distribution $p(\mathbf{Z}, \{\mathbf{F}^{(m)}\}|\mathbf{X}, \mathbf{Y})$ is intractable, so we will resort to approximate techniques.

3.1. Variational approximation

If the hyperparameters are known, it is possible to approximately compute the posterior using a variational approximation. We can use Jensen's inequality to construct a lower bound on the marginal likelihood as follows:

$$\log p(\mathbf{Y}|\mathbf{X}) = \log \int p(\mathbf{Y}|\{\mathbf{F}^{(m)}\}, \mathbf{Z}) p(\mathbf{Z}) \prod_{m=1}^{M} p(\mathbf{F}^{(m)}|\mathbf{X}) \mathrm{d}\{\mathbf{F}^{(m)}\}\mathrm{d}\mathbf{Z} \tag{6}$$

$$\geq \int q(\{\mathbf{F}^{(m)}\}, \mathbf{Z}) \log \frac{p(\mathbf{Y}|\{\mathbf{F}^{(m)}\}, \mathbf{Z}) p(\mathbf{Z}) \prod_{m=1}^{M} p(\mathbf{F}^{(m)})|\mathbf{X})}{q(\{\mathbf{F}^{(m)}\}, \mathbf{Z})} \mathrm{d}\{\mathbf{F}^{(m)}\}\mathrm{d}\mathbf{Z} = \mathcal{L}_{\mathrm{VB}}.$$

Here $\mathcal{L}_{\mathrm{VB}}$ is a lower bound on $\log p(\mathbf{Y}|\mathbf{X})$ for any *variational distribution* $q(\{\mathbf{F}^{(m)}\}, \mathbf{Z})$ and equality is attained if and only if $q(\{\mathbf{F}^{(m)}\}, \mathbf{Z}) = p(\mathbf{Z}, \{\mathbf{F}^{(m)}\}|\mathbf{X}, \mathbf{Y})$. Our objective is therefore to find a variational distribution that maximizes $\mathcal{L}_{\mathrm{VB}}$, and thus becomes an approximation to the true posterior. We will restrict our search to variational distributions that factorize as $q(\{\mathbf{F}^{(m)}\}, \mathbf{Z}) = q(\{\mathbf{F}^{(m)}\})q(\mathbf{Z})$.

This would increase the computational complexity of inference for this model, but the following derivations can still be applied.

If we assume that $q(\{\mathbf{F}^{(m)}\})$ is given (and therefore, also the marginals $q(\mathbf{f}_d^{(m)}) = \mathcal{N}(\mathbf{f}_d^{(m)}|\boldsymbol{\mu}_d^{(m)}, \boldsymbol{\Sigma}^{(m)})$ are available), it is possible to analytically maximize $\mathcal{L}_{\mathrm{VB}}$ with respect to $q(\mathbf{Z})$ by setting its derivative to zero and constraining it to be a probability density. The optimal $q(\mathbf{Z})$ is then:

$$q(\mathbf{Z}) = \prod_{n=1,m=1}^{N,M} [\hat{\boldsymbol{\Pi}}]_{nm}^{[\mathbf{Z}]_{nm}} \text{ with } [\hat{\boldsymbol{\Pi}}]_{nm} \propto [\boldsymbol{\Pi}]_{nm} \exp(a_{nm}) \tag{7}$$

$$\text{with} \qquad a_{nm} = \sum_{d=1}^{D} \left(-\frac{1}{2\sigma^2} \left(([\mathbf{y}_d]_n - [\boldsymbol{\mu}_d^{(m)}]_n)^2 + [\boldsymbol{\Sigma}^{(m)}]_{nn} \right) - \frac{1}{2} \log(2\pi\sigma^2) \right),$$

where we see that the (approximate) posterior distribution over the indicators $q(\mathbf{Z})$ factorizes for each sample.

Analogously, assuming $q(\mathbf{Z})$ as known, it is possible to analytically obtain the distribution over the latent functions that maximizes $\mathcal{L}_{\mathrm{VB}}$. For the OMGP model, this distribution factorizes both over trajectories and dimensions, and is given by

$$q(\mathbf{f}_d^{(m)}) = \mathcal{N}(\mathbf{f}_d^{(m)}|\boldsymbol{\mu}_d^{(m)}, \boldsymbol{\Sigma}^{(m)}) \tag{8a}$$

$$\text{with } \boldsymbol{\Sigma}^{(m)} = (\mathbf{K}^{-1(m)} + \mathbf{B}^{(m)})^{-1} \text{ and } \boldsymbol{\mu}_d^{(m)} = \boldsymbol{\Sigma}^{(m)} \mathbf{B}^{(m)} \mathbf{y}_d^{(m)} \tag{8b}$$

where $\mathbf{B}^{(m)}$ is a diagonal matrix with elements $[\hat{\boldsymbol{\Pi}}]_{1m}/\sigma^2 \ldots [\hat{\boldsymbol{\Pi}}]_{Nm}/\sigma^2$.

It is now possible to initialize $q(\mathbf{Z})$ and $q(\mathbf{f}_d^{(m)})$ from their prior distributions and iterate updates (7) and (8) to obtain increasingly refined approximations to the posterior. Since both steps are optimal with respect to the distribution that they compute, they are guaranteed to increase $\mathcal{L}_{\mathrm{VB}}$, and therefore the algorithm is guaranteed to converge to a local maximum.

Monotonous convergence can be monitored by computing $\mathcal{L}_{\mathrm{VB}}$ after each

update. $\mathcal{L}_{\mathrm{VB}}$ can be expressed as

$$\mathcal{L}_{\mathrm{VB}} = \left\langle \log p(\mathbf{Y}|\{\mathbf{F}^{(m)}\}, \mathbf{Z}) \right\rangle_{q(\{\mathbf{F}^{(m)}\},\mathbf{Z})} - \mathrm{KL}(q(\{\mathbf{F}^{(m)}\})||p(\{\mathbf{F}^{(m)}\})) - \mathrm{KL}(q(\mathbf{Z})||p(\mathbf{Z}))$$

where the first term is given by

$$\left\langle \log p(\mathbf{Y}|\{\mathbf{F}^{(m)}\}, \mathbf{Z}) \right\rangle_{q(\{\mathbf{F}^{(m)}\},\mathbf{Z})} = \sum_{n,m}^{N,M} [\hat{\mathbf{\Pi}}]_{nm} a_{nm} \ ,$$

and the two remaining terms are the Kullback-Leibler (KL) divergences from the approximate posterior to the prior, which are straightforward to compute.

Update (7) takes only $\mathcal{O}(NM)$ computation time, whereas (8) takes $\mathcal{O}(MN^3)$ time, due to the M matrix inversions. The presented model therefore has the same limitations as conventional GPs regarding the size of the data sets that it can be applied to. However, when the posterior probability of some indicator $[\hat{\mathbf{\Pi}}]_{nm}$ is close to zero, sample n no longer affects trajectory m and can be dropped in its computation, thus reducing the cost. Furthermore, it is possible to use sparse GPs[4] to reduce this cost[5] to $\mathcal{O}(MN)$ time by making use of the matrix inversion lemma.

3.2. An improved variational bound for OMGP

So far we have assumed that all the hyperparameters of the model are known. However, in practice, some procedure to select them is needed. The

[4]Such as the standard FITC approximation, described in [22] or the variational approach introduced in [23].

[5]Obviously, the cost also depends on the quality of the approximation by a constant factor. If the FITC approximation with r pseudo-inputs (or other rank-r approximation) is used, the computational complexity could be expressed as $\mathcal{O}(MNr^2)$.

most straightforward way of achieving this would be to select them so as to maximize $\mathcal{L}_{\mathrm{VB}}$, interleaving this procedure with updates (7) and (8). However, when the quality of this bound is sensitive to changes of the model hyperparameters, this approach results in very slow convergence. A solution to this problem is described in [18] where the advantages of maximizing an alternative, tighter bound on the likelihood are shown.

The improved bound proposed in [18] is still a lower bound on the likelihood but it can be proved that it is also an upper bound on the standard variational bound $\mathcal{L}_{\mathrm{VB}}$. As shown in [18], if we subtract $\mathcal{L}_{\mathrm{VB}}$ from the improved bound, the result takes on the form of a KL-divergence. This fact can be used both to show that it upper-bounds $\mathcal{L}_{\mathrm{VB}}$ (since KL-divergences are always positive) and to name the new bound, which is referred to as the KL-corrected variational bound.

The KL-corrected bound for the OMGP model arises when the term $\log \int p(\mathbf{Y}|\{\mathbf{F}^{(m)}\}, \mathbf{Z})p(\mathbf{Z})\mathrm{d}\mathbf{Z}$ from the true marginal likelihood (6) is replaced with $\int q(\mathbf{Z}) \log \frac{p(\mathbf{Y}|\{\mathbf{F}^{(m)}\},\mathbf{Z})p(\mathbf{Z})}{q(\mathbf{Z})}\mathrm{d}\mathbf{Z}$, which according to Jensen's inequality, constitutes a lower bound for any distribution $q(\mathbf{Z})$:

$$\log p(\mathbf{Y}|\mathbf{X}) \geq \log \int \prod_{m=1}^{M} p(\mathbf{F}^{(m)}|\mathbf{X}) e^{\int q(\mathbf{Z}) \log \frac{p(\mathbf{Y}|\{\mathbf{F}^{(m)}\},\mathbf{Z})p(\mathbf{Z})}{q(\mathbf{Z})}\mathrm{d}\mathbf{Z}} \mathrm{d}\{\mathbf{F}^{(m)}\} =$$

$$\mathcal{L}_{\mathrm{CorrVB}} = \sum_{m=1,d=1}^{M,D} \log \mathcal{N}(\mathbf{y}_d^{(m)}|\mathbf{0}, \mathbf{K}^{(m)} + \mathbf{B}^{-1(m)})$$

$$- \mathrm{KL}(q(\mathbf{Z})||p(\mathbf{Z})) + \frac{D}{2} \sum_{n=1,m=1}^{N,M} \log \frac{(2\pi\sigma^2)^{1-[\hat{\mathbf{\Pi}}]_{nm}}}{[\hat{\mathbf{\Pi}}]_{nm}} \,.$$

The KL-corrected lower bound $\mathcal{L}_{\mathrm{CorrVB}}$ can be computed analytically and has the advantage with respect to $\mathcal{L}_{\mathrm{VB}}$, of depending only on $q(\mathbf{Z})$ (and not

$q(\{\mathbf{F}^{(m)}\}))$, since it is possible to integrate $\prod_{m=1}^{M} p(\mathbf{F}^{(m)}|\mathbf{X})$ out analytically.

Bound $\mathcal{L}_{\text{CorrVB}}$ can be alternatively obtained by following the recent work in [19] and optimally removing $q(\{\mathbf{F}^{(m)}\})$ from the standard bound. In the context of that work, $\mathcal{L}_{\text{CorrVB}}$ is referred to as the "marginalized variational bound", and it is made clear that $\mathcal{L}_{\text{CorrVB}}$ corresponds simply to $\mathcal{L}_{\text{VB}}$ when, for a given $q(\mathbf{Z})$, the optimal choice for $q(\{\mathbf{F}^{(m)}\})$ is made. In other words, for the same set of hyperparameters and the same $q(\mathbf{Z})$, if one choses $q(\{\mathbf{F}^{(m)}\})$ according to (8), both $\mathcal{L}_{\text{VB}}$ and $\mathcal{L}_{\text{CorrVB}}$ would provide the same result.

Thus, learning is performed simply by optimizing $\mathcal{L}_{\text{CorrVB}}$ with respect to $q(\mathbf{Z})$ and the hyperparameters, iterating the following two steps:

- E-Step: Updates (7) and (8) are alternated, which monotonically increase both $\mathcal{L}_{\text{VB}}$ and $\mathcal{L}_{\text{CorrVB}}$, until convergence. Hyperparameters are kept fixed.

- M-Step: Gradient descent of $\mathcal{L}_{\text{CorrVB}}$ with respect to all hyperparameters is performed. Distribution $q(\mathbf{Z})$ is kept fixed.

Note that it is in the M-step where $\mathcal{L}_{\text{CorrVB}}$ becomes actually useful, since this improved bound remains more stable across different hyperparameter selections, due to it not depending on $q(\{\mathbf{F}^{(m)}\})$, as demonstrated in [18].

Of course, any strategy that maximizes $\mathcal{L}_{\text{CorrVB}}$ is valid, but we have found the above EM procedure to work well in practice.

Computing $\mathcal{L}_{\text{CorrVB}}$ according to the provided expression without incurring in numerical errors can be challenging in practice, since several inversions, which maybe unstable, are needed. Also, note that $\mathbf{B}^{(m)}$ can take arbitrarily small values and thus direct inversion may not be possible. An

implementation-friendly expression where explicit inverses are avoided is

$$\mathcal{L}_{\text{CorrVB}} = \sum_{m=1}^{M} \Big(-\frac{1}{2} \sum_{d=1}^{D} ||\mathbf{R}^{(m)\top} \backslash (\mathbf{B}^{(m)\frac{1}{2}} \mathbf{y}_d^{(m)})||^2 - D \sum_{n=1}^{N} \log[\mathbf{R}^{(m)}]_{nn} \Big)$$
$$- \text{KL}(q(\mathbf{Z})||p(\mathbf{Z})) - \frac{D}{2} \sum_{n=1,m=1}^{N,M} [\hat{\mathbf{\Pi}}]_{nm} \log(2\pi\sigma^2) \ ,$$

where

$$\mathbf{R}^{(m)} = \text{chol}(\mathbf{I} + \mathbf{B}^{(m)\frac{1}{2}} \mathbf{K}^{(m)} \mathbf{B}^{(m)\frac{1}{2}})$$

and the backslash has the usual meaning of solution to a linear system.[6]

3.3. Predictive distributions

The OMGP model can be used for a variety of tasks. In the data association problem (i.e., clustering data into trajectories) the task at hand is to cluster observations into trajectories, which can be achieved by assigning each observation to the trajectory that more likely generated it, i.e., to assign label $m^* = \arg\max_m [\hat{\mathbf{\Pi}}]_{nm}$ to the n-th observation, so no further computations are necessary. For other tasks, however, it can be necessary to obtain predictive distributions over the output space at new locations. Under the variational approximation, this predictive distributions can be computed analytically.

The predictive distribution in the output dimension d corresponding to a

[6]Expressions of the type $\mathbf{C}\backslash\mathbf{c}$ refer to the solution of the linear system $\mathbf{C}\mathbf{x} = \mathbf{c}$ and are a numerically stable operation requiring only $\mathcal{O}(N^2)$ time when $\mathbf{C}$ is triangular, which is the case here.

new test input location $\mathbf{x}_*$ can be expressed as

$$
\begin{aligned}
p(y_{*d}|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) &= \sum_{m=1}^{M} [\mathbf{\Pi}]_{*m} \int p\ (y_{*d}|\mathbf{f}_d^{(m)}, \mathbf{x}_*, \mathbf{X})\ \ p\ (\mathbf{f}_d^{(m)}|\mathbf{X}, \mathbf{Y}) \mathrm{d}\mathbf{f}_d^{(m)} \\
&\approx \sum_{m=1}^{M} [\mathbf{\Pi}]_{*m} \int p\ (y_{*d}|\mathbf{f}_d^{(m)}, \mathbf{x}_*, \mathbf{X})\ \ q\ (\mathbf{f}_d^{(m)}|\mathbf{X}, \mathbf{Y}) \mathrm{d}\mathbf{f}_d^{(m)} \\
&= \sum_{m=1}^{M} [\mathbf{\Pi}]_{*m} \mathcal{N}(y_{*d}|\mu_{*d}^{(m)}, \sigma_{*d}^{2(m)})
\end{aligned}
$$

with

$$
\begin{aligned}
\mu_{*d}^{(m)} &= \mathbf{k}_*^{\top(m)}\ (\mathbf{K}^{(m)} + \mathbf{B}^{(m)-1})^{-1}\ \mathbf{y}_d, \\
\sigma_{*d}^{2(m)} &= \sigma^2 + k_{**} - \mathbf{k}_*^{\top(m)}\ (\mathbf{K}^{(m)} + \mathbf{B}^{(m)-1})^{-1}\ \mathbf{k}_*^{(m)}\ ,
\end{aligned}
$$

i.e., a Gaussian mixture under the approximate posterior. The mixing factors $[\mathbf{\Pi}]_{*m}$ are the prior probabilities of each component, one of the given hyperparameters of the model, and typically constant for all inputs.

Note the correspondence of these predictive equations with the standard predictions for GP regression (2). The only difference is the noise component, which is scaled for each sample according to $[\hat{\mathbf{\Pi}}]_{nm}^{-1}$. In particular, as the posterior probability of a sample belonging to the current trajectory (sometimes known as "responsibility") decays, the amount of noise associated to that sample is proportionally grown, thus reducing its effect on the posterior process.

Due to the reasons mentioned in the previous subsection, the predictive equations should not be implemented directly. Instead, the following

numerically-stable expressions should be used:

$$\mu_{*d}^{(m)} = \mathbf{k}_*^{\top(m)} \ \mathbf{B}^{(m)\frac{1}{2}}(\mathbf{R}^{(m)}\backslash(\mathbf{R}^{(m)\top}\backslash(\mathbf{B}^{(m)\frac{1}{2}}\mathbf{y}_d^{(m)}))),$$
$$\sigma_{*d}^{2(m)} = \sigma^2 + k_{**} - ||\mathbf{R}^{(m)\top}\backslash(\mathbf{B}^{(m)\frac{1}{2}}\mathbf{k}_*^{\top(m)})||^2 \ .$$

3.4. Batch versus online operation

Though the description of OMGP is oriented towards batch data association tasks, this model can also be successfully applied to online tasks, by using a data set that grows over time. New samples are included as they arrive and the learning process is re-started, initializing it from the state that was obtained as a solution for the previous problem. Depending on the constraints of a given problem, many different optimizations can be made to avoid an explosion in computational effort, such as using low-rank updates.

Note, however, that since in this model all the elements in each latent function form a fully connected graph, the Markovian property does not hold and the computation time required for each update is not constant. A possible workaround to achieve constant-time updates is to use constat-size data sets, for instance corresponding to a sliding window, and then perform low-rank updates to include and remove samples. However, we will not pursue that option in this work.

4. Experiments

In this section we investigate the behavior of OMGP both in data association tasks and regression tasks, showing the versatility of this model. We use an implementation of OMGP in Matlab on a 3GHz, dual-core desktop PC with 4GB of memory, yielding executions times of the order of seconds for each experiment.

4.1. Data association tasks

4.1.1. Toy data

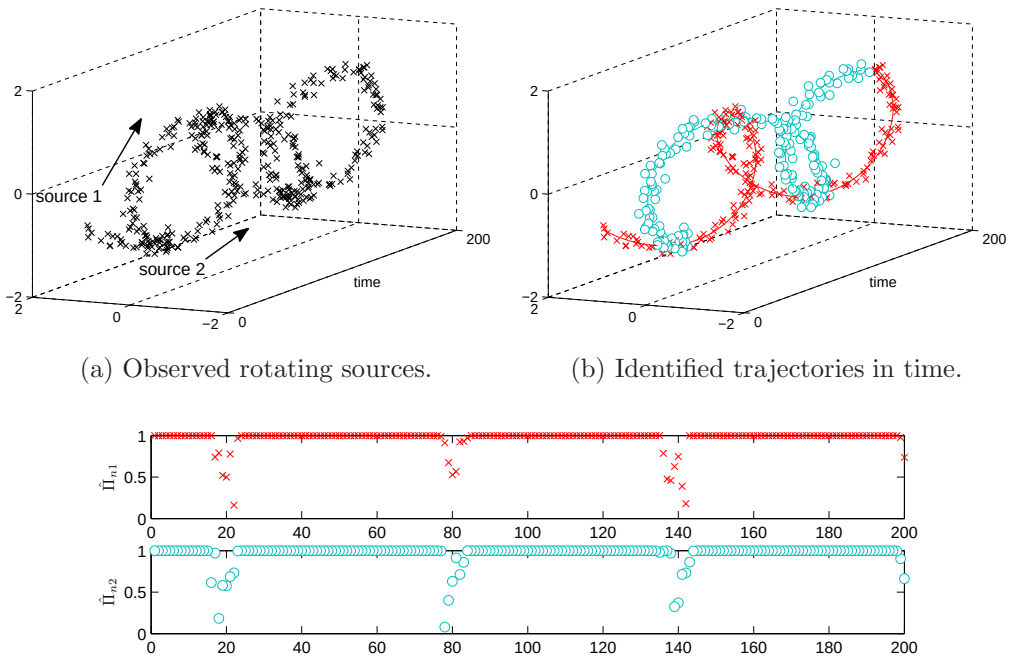
We first apply OMGP to perform data association on a toy data set. The sources perform circular motions, one clockwise and one counterclockwise, as depicted in Fig. 2(a). The available observations represent the measured positions of the sources (which include Gaussian noise) at known time instants. However, it is not known which observed position corresponds to which source. Since both trajectories are circles with the same center and radius, the sources cross each other twice per revolution, making the clustering problem more difficult. However, as shown in Fig. 2(b), OMGP is capable of successfully identifying the unknown trajectories. Fig. 2(c) illustrates the uncertainty about the estimated labels. Specifically, it shows a decrease in the posterior probability of the correct labels whenever the two sources come close.

4.1.2. Missile-to-air multi-target tracking

Next, we consider a missile-to-air tracking scenario as described in [7]. The motion dynamics of this scenario are defined by the following state-space equations:

$$\mathbf{s}_{t+1} = \begin{bmatrix} \mathbf{I}_3 & T\mathbf{I}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{s}_t + \begin{bmatrix} \frac{T^2}{2}\mathbf{I}_3 \\ T\mathbf{I}_3 \end{bmatrix} \mathbf{v}_t; \quad \mathbf{r}_t = h(\mathbf{s}_t) = \begin{bmatrix} \sqrt{X_t^2 + Y_t^2 + Z_t^2} \\ \arctan(\frac{Y_t}{X_t}) \\ \arctan(\frac{-Z_t}{\sqrt{X_t^2+Y_t^2}}) \end{bmatrix} + \mathbf{e}_t.$$

In this model, the state vector $\mathbf{s}_t = [X_t, Y_t, Z_t, V_{x,t}, V_{y,t}, V_{z,t}]$ contains the source position and velocity components, $\mathbf{r}_t$ contains the observed measurements, T is the sampling interval, and $\mathbf{I}_3$ and $\mathbf{O}_3$ represent the 3×3 unity



(a) Observed rotating sources.

(b) Identified trajectories in time.

(c) Posterior probability of correct labels.

Figure 2: (a) Observations for two sources that move in opposite circles. (b) The data association solution obtained by OMGP. (c) Posterior probability of the correct label for observations coming from source 1 (top) and 2 (bottom).

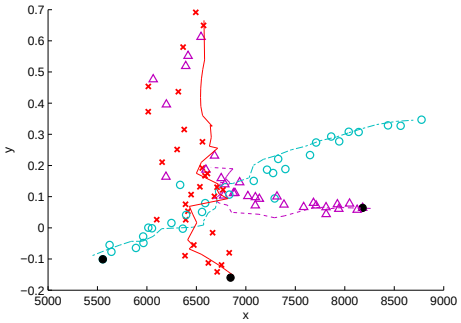
matrix and null matrix, respectively. The process noise $\mathbf{v}_t$ and measurement noise $\mathbf{e}_t$ are assumed Gaussian, $\mathbf{v}_t \in \mathcal{N}(0, \mathbf{Q})$ and $\mathbf{e}_t \in \mathcal{N}(0, \mathbf{R})$. For more details refer to [7]. The problem posed in [7] consists in tracking two sources and estimating their unknown state vector, given their correct initial states $\mathbf{s}_0^1 = [6500, -1000, 2000, -50, 100, 0]$ and $\mathbf{s}_0^2 = [5050, -450, 2000, 100, 50, 0]$. We consider a more complex scenario by adding a third source, with initial state $\mathbf{s}_0^3 = [8000, 500, 2000, -100, 0, 0]$, which passes close to one of the other sources at a certain instant.

We apply the SIR/MCJPDA filter from [7] and OMGP to perform data

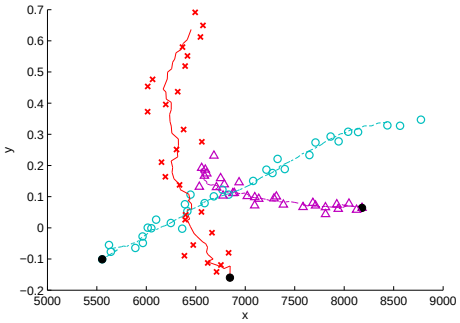
association on the observations. The SIR/MCJPDA filter consists of a set of joint particle filters that perform tracking of multiple sources, combined with a joint probability data association (JPDA) technique which provides instantaneous data association. The number of particles used in this experiment is 25000. In order to operate correctly, the SIR/MCJPDA filter requires complete knowledge of the used state-space model and the initial state vectors $\mathbf{x}_0^i$. Note that OMGP is completely blind in this regard. The OMGP algorithm is operated first in its incremental online setting. For illustration purposes, we also include results of the batch version of the OMGP algorithm.

The trajectories obtained by each method can be found in Fig. 3, along with the predicted measurements. Although the SIR/MCJPDA filter initially performs correctly, it encounters difficulties at the point where the sources come close. After this point it shows erroneous assignments for at least one trajectory. Its mistakes are mainly due to its state vector depending only on 1 previous state, which proves insufficient if the sources are close during multiple consecutive measurements. The online version of OMGP does not show this problem. The smoothest solution is obtained by batch OMGP, which performs a global evaluation of the entire trajectories.

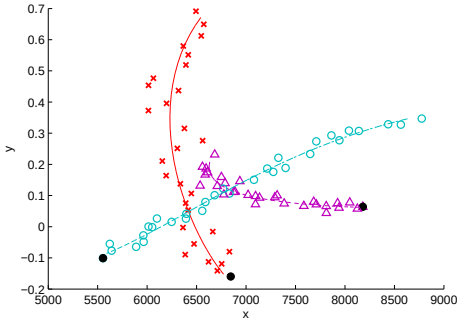
To evaluate the performance of the algorithms, we measure the RMSE of each trajectory. These values can be found in Table 1, along with the number of observations that are assigned to the wrong trajectories, n_{err}, out of a total of 90 observations. As can be observed, both versions of the OMGP algorithm obtain superior results compared to SIR/MCJPA. Furthermore, while SIR/MCJPDA requires complete knowledge of the state-space model and the initial state vectors, OMGP does not require any knowledge of the



(a) Trajectories identified by SIR/MCJPDA.



(b) Trajectories identified by OMGP, incremental online version.



(c) Trajectories identified by OMGP, batch solution.

Figure 3: Missile-to-air data association problem with three sources. The starting point of each source is marked with a black dot.

underlying model.

4.1.3. Interference alignment in OFDM wireless networks

Interestingly, the data association problem can be found in contexts that go beyond standard multi-target tracking scenarios, such as digital communications [24]. In the third experiment we apply OMGP to a data association problem that occurs in wireless communication networks.

Table 1: RMSE comparison on the missile-to-air data association problem.

Algorithm	RMSE #1	RMSE #2	RMSE #3	n_{err}
SIR/MCJPDA	292.46	150.07	258.14	17
OMGP (online)	182.31	151.46	163.92	6
OMGP (batch)	133.30	80.23	118.94	1

Interference alignment (IA) is a concept that has recently emerged as a solution to raise the capacity of wireless multiple-input multiple-output (MIMO) networks [25]. The underlying idea of IA along the spatial dimensions is that the interference from other transmitters must be aligned at each receiver in a subspace orthogonal to the signal space. In order to implement interference alignment in scenarios with multiple subcarriers, a digital filter must be applied at each transmit antenna. Here we will consider a 3-user interference channel with two antennas per node and OFDM modulation using N_c subcarriers [26], which allows for two possible filter responses per subcarrier. Since only smooth frequency responses can be implemented, the smoothest solution of the 2^{N_c} possible choices should be selected.

This combinatorial problem corresponds to a data association problem in which only the smoothest curve is of interest. (see Fig. 4(a)). The data used for this experiment consists of two simulated data sets and one data set obtained with a MIMO test bed setup[7], each using 52 subcarriers. In Fig. 4 we illustrate the solutions obtained by OMGP on these data sets. While

[7]See [27] for a full description of the used test bed.

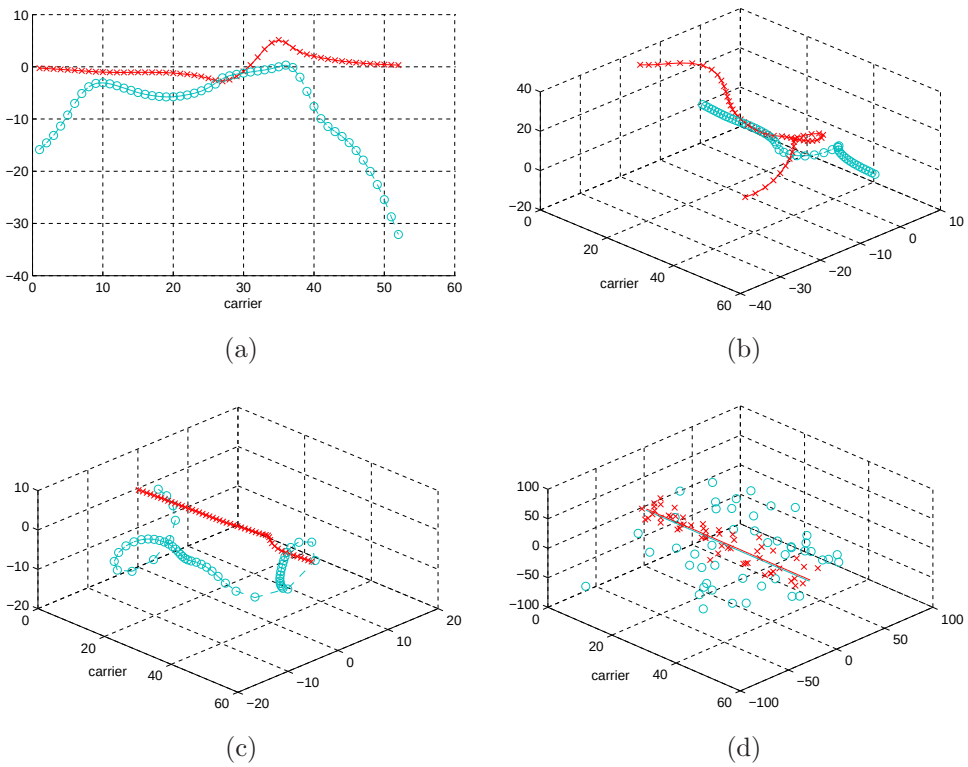


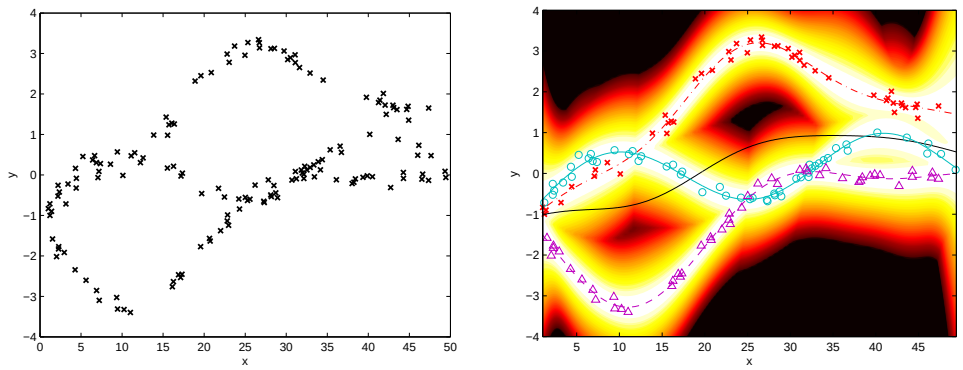
Figure 4: Data association results obtained by OMGP on different interference alignment problems. (a) shows the IA solutions (imaginary part only) for the first simulated data set. (b) and (c) show the solutions for the first and second simulated data sets (real and imaginary part versus subcarrier number). (d) shows the IA solution for a real-world data set. Note that complex values are simply simply treated as two-dimensional real data in this experiment.

the simulated data sets from Fig. 4(b) and Fig. 4(c) represent reasonably simple data association problems, the performance of OMGP on the real-world data set of Fig. 4(d) shows that it is capable of correctly distinguishing the smoothly-varying solution from the surrounding noisy data. As a matter of fact, we have been able to successfully implement OMGP in the IA setting

for a parallel ungoing research project.

4.2. Regression tasks

We now consider application of the model in more standard regression tasks. In particular, we consider tasks where the target density is multimodal, which is the case when the data comes from multiple sources.



(a) Original data set.

(b) Inferred labels and predictive log-probs.

Figure 5: Posterior log-probability of the OMGP model and label inference.

4.2.1. Multilevel regression

Consider the data set from Fig. 5(a), which corresponds to observations from three independent functions. A normal GP would fail to produce valid multimodal outputs and previously proposed mixtures of GPs would restrict the component GPs to local parts of the space. OMGP can properly label each observation according to the generating function and provide multimodal predictive distributions, as depicted in Fig. 5(b).

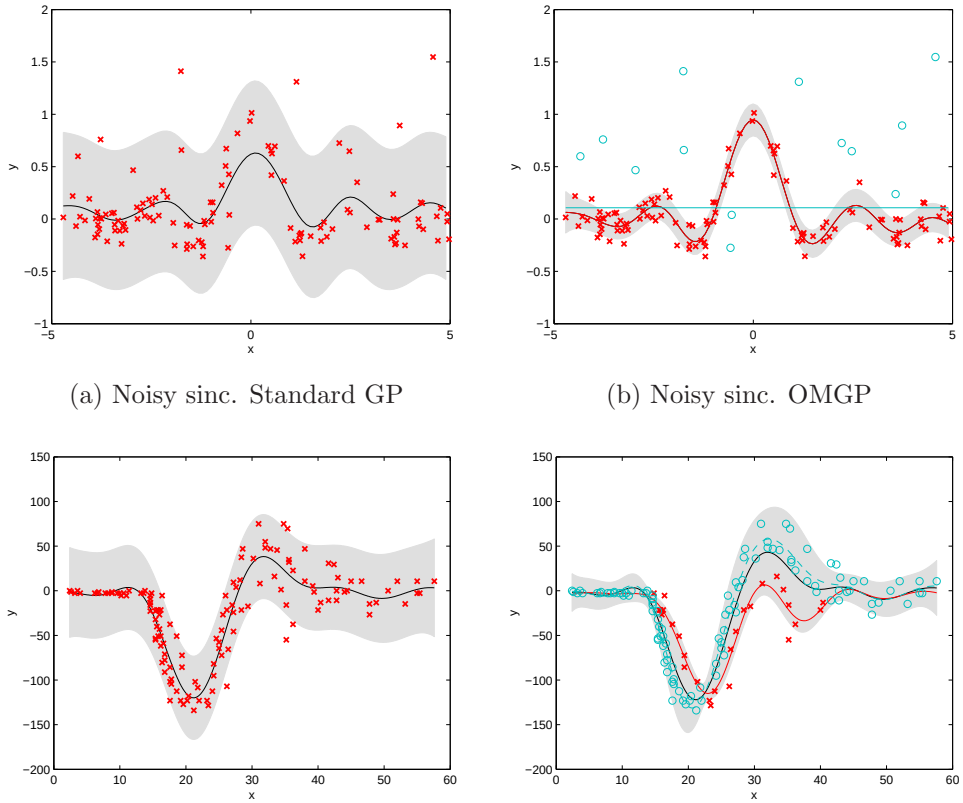
Fig. 5 can also be interpreted as measurements of the position of three particles moving along one dimension, of which snapshots are taken at irregular time intervals (horizontal axis). Each snapshot introduces noise in the position measurement and does not necessarily capture the position of all the particles. In this case OMGP could be used to predict the position of any particle at any given point in time, as well as to properly label the samples in each snapshot.

4.2.2. Robust regression

Since each GP in the mixture can use a different covariance function, it is possible to use a GP to capture unrelated outliers and another one to interpolate the main function. This is easily achieved by a mixture of two GPs, one with the ARD-SE covariance function and another with $k(x, x') = b^2\delta(x, x')$, i.e., white noise. We consider the problem of regression in a noisy sinc in which some outliers have been introduced in Fig. 6 (top row). Observe how OMGP both identifies the outliers and ignores them, resulting in much better predictive means and variances.

4.2.3. Heteroscedastic behavior

Finally, Fig. 6 (bottom row) shows the results of running a GP and OMGP on the motorcycle data set from [28]. Two components have been identified, which might or might not correspond to two actual physical mechanisms alternatively producing observations. The predictive variances show improved behavior with respect to the standard GP.



(a) Noisy sinc. Standard GP

(b) Noisy sinc. OMGP

(c) Motorcycle. Standard GP

(d) Motorcycle. OMGP

Figure 6: Predictive means and variances for two different data sets. The shaded area denotes ± 2 standard deviations around the mean. Top row: Noisy sinc with outliers. (a) Standard GP and (b) OMGP with a noise-only component. (Only the predictive mean and variance of the signal component is depicted, which includes noise σ^2). Bottom row: Silverman's motorcycle data set.

5. Discussion and future work

In this work we have introduced a novel GP mixture model inspired by multi-target tracking problems. The new model has the important difference with respect to previous approaches of using global mixture components and assigning samples to components by relying on their value in output space,

instead of input space (as it is done when gating functions are used).

A simple and efficient algorithm for inference relying on the variational Bayesian framework has been provided. The model can be applied in practice due to the use of an improved, KL-corrected variational bound to learn the hyperparameters. Direct optimization of this bound both to obtain an approximate posterior and to learn the hyperparameters will be considered in a further work.

The OMGP model offers promising results when tracking moving targets, as has been illustrated experimentally in Section 4 and compares favorably with established methods in the field. Also, through imaginative application of the model using different covariance functions we were able to adapt the approach to robust regression and heteroscedastic noise.

Naive implementation of GPs limits their applicability to only a few thousand data samples. However, recent advances in sparse approximations (e.g. [22, 23]) greatly should enable our approach to be applied to much larger data sets.

6. Acknowledgments

The authors wish to thank Oscar González, University of Cantabria, for providing the data used in the interference alignment experiment. The first and second authors were supported by MICINN (Spanish Ministry for Science and Innovation) under grants TEC2010-19545-C04-03 (COSIMA) and CONSOLIDER-INGENIO 2010 CSD2008-00010 (COMONSENS). Additionally, funding to support part of this collaborative effort was provided by PASCAL's Internal Visiting Programme.

[1] Y. Bar-Shalom, Tracking and data association, Academic Press Professional, Inc. San Diego, CA, USA, 1987.

[2] I. J. Cox, A review of statistical data association techniques for motion correspondence, International Journal of Computer Vision 10 (1993) 53–66.

[3] S. Ullman, The interpretation of visual motion, M.I.T. Press, Cambridge, MA, USA, 1979.

[4] J. Singh, U. Madhow, S. Suri, R. Cagley, Multiple target tracking with binary proximity sensors, ACM Transactions on sensor networks (accepted for publication).

[5] T. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, IEEE Journal of Oceanic Engineering 8 (1983) 173 – 184.

[6] D. Reid, An algorithm for tracking multiple targets, Automatic Control, IEEE Transactions on 24 (1979) 843 – 854.

[7] R. Karlsson, F. Gustafsson, Monte Carlo data association for multiple target tracking, IEEE International Seminar on Target Tracking: Algorithms and Applications 1 (2001) 13.

[8] D. Chetverikov, J. Verestói, Feature point tracking for incomplete trajectories, Computing 62 (1999) 321–338.

[9] C. Veenman, M. Reinders, E. Backer, Resolving motion correspondence

for densely moving points, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 54–72.

[10] V. Nagarajan, M. Chidambara, R. Sharma, Combinatorial problems in multitarget tracking - a comprehensive solution, IEE Proceedings-F: Communications, Radar and Signal Processing 134 (1987) 113 –118.

[11] I. Cox, S. Hingorani, An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 138 –150.

[12] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[13] V. Tresp, A Bayesian committee machine, Neural Computation 12 (2000) 2719–2741.

[14] C. E. Rasmussen, Z. Ghahramani, Infinite mixtures of Gaussian process experts, in: Advances in Neural Information Processing Systems 14, MIT Press, 2002, pp. 881–888.

[15] E. Meeds, S. Osindero, An alternative infinite mixture of Gaussian process experts, in: Advances in Neural Information Processing Systems 18, MIT Press, 2006, pp. 883–890.

[16] C. Yuan, C. Neubauer, Variational mixture of Gaussian process experts, in: Advances in Neural Information Processing Systems 21, 2009, pp. 1897–1904.

[17] C. Tay, C. Laugier, Modelling smooth paths using Gaussian processes, in: International Conference on Field and Service Robotics, 2007, pp. 381–390.

[18] N. J. King, N. Lawrence, Fast variational inference for Gaussian process models through KL-correction, in: ECML, Lecture Notes in Computer Science, Berlin, 2006, pp. 270–281.

[19] M. Lázaro-Gredilla, M. Titsias, Variational heteroscedastic Gaussian process regression, in: 28th International Conference on Machine Learning, Omnipress, Bellevue, WA, USA, 2011, pp. 841–848.

[20] C. E. Rasmussen, Evaluation of Gaussian Processes and other Methods for Non-linear Regression, Ph.D. thesis, University of Toronto, 1996.

[21] E. V. Bonilla, K. M. A. Chai, C. K. I. Williams, Multi-task Gaussian process prediction, in: Advances Neural Information Processing Systems 20, pp. 153–160.

[22] E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in: Advances in Neural Information Processing Systems 18, MIT Press, 2006, pp. 1259–1266.

[23] M. K. Titsias, Variational learning of inducing variables in sparse Gaussian processes, in: Proceedings of the 12th International Workshop on AI Stats, pp. 567–574.

[24] S. Van Vaerenbergh, I. Santamaria, P. Barbano, U. Ozertem, D. Erdogmus, Path-based spectral clustering for decoding fast time-varying

MIMO channels, in: IEEE International Workshop on Machine Learning for Signal Processing, IEEE, pp. 1–6.

[25] V. R. Cadambe, S. A. Jafar, Interference alignment and degrees of freedom of the K-user interference channel, IEEE Transactions on Information Theory 54 (2008) 3425 –3441.

[26] J. Proakis, Digital Communications, McGraw-Hill, 1995.

[27] J. Gutiérrez, Ó. González, J. Pérez, D. Ramírez, L. Vielva, J. Ibáñez, I. Santamaría, Frequency-domain methodology for measuring MIMO channels using a generic test bed, IEEE Transactions on Instrumentation and Measurement 60 (2011) 827–838.

[28] B. W. Silverman, Some aspects of the spline smoothing approach to non-parametric regression curve fitting, Journal of the Royal Statistical Society 47 (1985) 1–52.