VARIATIONAL GAUSSIAN PROCESS FOR SENSOR FUSION

Neda Rohani[1⋆], *Pablo Ruiz*[2∗], *Emre Besler*[1⋆], *Rafael Molina*[2∗] *and Aggelos K. Katsaggelos*[1+]

[1] Dpt. of Electrical Engineering and Computer Science. Northwestern University, USA.
[2] Dpto. de Ciencias de la Computación e I.A. Universidad de Granada, Spain.

e-mail: ⋆{nedarohani2019,emrebesler2020}@u.northwestern.edu,∗{mataran,rms}@decsai.ugr.es, +aggk@eecs.northwestern.edu

ABSTRACT

In this paper, we introduce a new Gaussian Process (GP) classification method for multisensory data. The proposed approach can deal with noisy and missing data. It is also capable of estimating the contribution of each sensor towards the classification task. We use Bayesian modeling to build a GP-based classifier which combines the information provided by all sensors and approximates the posterior distribution of the GP using variational Bayesian inference. During its training phase, the algorithm estimates each sensor's weight and then uses this information to assign a label to each new sample. In the experimental section, we evaluate the classification performance of the proposed method on both synthetic and real data and show its applicability to different scenarios.

Index Terms— Gaussian process, fusion, Bayesian modeling, variational inference, classification.

1. INTRODUCTION

Fusing information from several sensors to perform a classification task is a widely applicable research topic within the Machine Learning community. It has two main benefits. Firstly, it can boost the classification performance of a single sensor by providing extra information on the classification task, and secondly, and even more importantly here, it compensates for noisy sensors or sensors with missing data. In other words, fusion brings robustness to noisy channels or partial loss of data.

It could be claimed that GPs [1] are one of the most preferred fusion tools. Cohn and Specia [2] utilize GPs for Natural Language Processing. They use multi-task learning to classify the quality of a sentence while determining the labels with crowdsourcing. They also model correlation between different annotators. Fox and Dunson [3] apply GPs to time series problems. They introduce multi-resolution GPs and obtain long-range dependencies, see also [4]. Duvenaud et al. [5] use an additive GP that generalizes Generalized Additive Models (GAM) and Squared-Exponential Gaussian Processes (SE-GP), two of the most commonly used regression models in statistics. Gerardo-Castro et al. [6] use GPs for robot sensing and vision. Girolami [7] proposes a GP model and applies it to biological data that overcomes the classification obstacles with SVM when dealing with a heterogeneous dataset. Kapoor et al. [8] use a GP fusion method to deal with missing channels or labeling noise in the data. Rather than omitting the noisy channel all together, the developed algorithm compensates for it via fusion. Rodriguez et al. [9] apply GPs to crowdsourcing problems aiming at making labeling from multiple annotators closer to the golden standard.

In this work, we propose a new modeling of the fusion classification problem using GPs (two class problem). This approach is capable of also working with missing samples and estimates the reliability of sensors. This idea was suggested in [10]. During the training phase, based on the each sensor reliability, the corresponding weights are estimated. These weights are used in order to classify a new point in the testing phase. We present the classification accuracy of the algorithm applied to both synthetic and real data sets. The rest of paper is organized as follows. In sections 2, our modeling of the problem is described. Variational Inference and the proposed algorithm to estimate the posterior distribution are presented in section 3. The classification rule is provided in section 4 where we also explain how to deal with missing data. Experimental results are presented in section 5 and finally section 6 concludes the paper.

2. BAYESIAN MODELING

Let us assume that we have a training set of N samples acquired using P different sensors. $\mathbf{x}_{ij} \in \mathbb{R}^{D_i \times 1}$ represents the j-th training sample, acquired by the i-th sensor, and the respective labels are given by the vector $\mathbf{y} = [y_1, \ldots, y_N]^T \in \{0,1\}^{N\times 1}$ (binary classification). We write

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \ldots & \mathbf{x}_{1N} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \ldots & \mathbf{x}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{P1} & \mathbf{x}_{P2} & \ldots & \mathbf{x}_{PN} \end{bmatrix} \in \mathbb{R}^{(D_1+D_2+\ldots+D_P)\times N}.$$

In this notation, a single sample of sensor i has dimension D_i. For the time being, we assume that all $\mathbf{x}_{ij}$ are available, that is, all samples are provided by all sensors. We will later

This work has been supported in part by the Department of Energy grant DE-NA0002520 and the Ministerio de Economía y Competitividad under contract TIN2013-43880-R.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

examine how to adapt the model to the case where some sensors do not provide information for some samples.

We assume that $\mathbf{f}_{i,\cdot} = (f_{i1}, \ldots, f_{iN})$, which is associated to the $i-$th sensor, follows a $\mathcal{N}(\mathbf{0}, \alpha_i \mathbf{K}_i)$, where $\mathbf{K}_i(p,q) = k_i(\mathbf{x}_{ip}, \mathbf{x}_{iq})$, k_i is a kernel function, and α_i is a variance parameter. Assuming that the sensors are independent, we then assign the following prior model to $\mathbf{f} = (\mathbf{f}_{1,\cdot}^T, \ldots, \mathbf{f}_{P,\cdot}^T)^T \in \mathbb{R}^{PN\times 1}$

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{1}$$

where $\mathbf{K}$ is a $PN \times PN$ matrix defined by blocks $(\mathbf{K}_i)$ as

$$\mathbf{K} = \text{diag}(\alpha_1 \mathbf{K}_1, \ldots, \alpha_P \mathbf{K}_P). \tag{2}$$

Then, the realizations of the Gaussian process and the observed labels are linked by the observation model

$$\begin{aligned} \mathrm{p}(\mathbf{y}|\mathbf{f}) &= \prod_{j=1}^{N} \left(\boldsymbol{\sigma}\left(\sum_{i=1}^{P} f_{ij}\right)\right)^{y_j} \left(\boldsymbol{\sigma}\left(-\sum_{i=1}^{P} f_{ij}\right)\right)^{1-y_j} \\ &= \prod_{j=1}^{N} \left(\boldsymbol{\sigma}\left(\mathbf{1}^T \mathbf{f}_{\cdot,j}\right)\right)^{y_j} \left(\boldsymbol{\sigma}\left(-\mathbf{1}^T \mathbf{f}_{\cdot,j}\right)\right)^{1-y_j}, \end{aligned} \tag{3}$$

where $\boldsymbol{\sigma}$ is the sigmoid function, $\boldsymbol{\sigma}(a) = 1/(1+\exp(-a))$, and $\mathbf{1}$ is a column vector of size P with all its components equal to one.

The joint distribution can be calculated as

$$\mathrm{p}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha}) = \mathrm{p}(\mathbf{y}|\mathbf{f})\mathrm{p}(\mathbf{f}|\mathbf{X}, \boldsymbol{\alpha})\mathrm{p}(\boldsymbol{\alpha}), \tag{4}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_P)$ and we use an improper flat prior for $\boldsymbol{\alpha}$. The rationale behind the proposed model is the following: each sensor is capable of independently providing a classifier from all the information it gathers. For a given sample j, adding the GP values associated with sensors i and i', f_{ij} and $f_{i'j}$, respectively, will increase (decrease) the likelihood of the observed label if they are in agreement (disagreement) in their signs.

3. VARIATIONAL INFERENCE

In a Bayesian framework, the inference on the unknowns is performed on the posterior distribution. In our case, the calculation of $\mathrm{p}(\mathbf{f}, \boldsymbol{\alpha}|\mathbf{y})$ is intractable, and therefore we approximate the posterior distribution by minimizing the Kulback-Leibler (KL) divergence

$$\begin{aligned} &\mathrm{KL}(\mathrm{q}(\mathbf{f}, \boldsymbol{\alpha})||\mathrm{p}(\mathbf{f}, \boldsymbol{\alpha}|\mathbf{y})) = \\ &\int_{\mathbf{f},\boldsymbol{\alpha}} \mathrm{q}(\mathbf{f}, \boldsymbol{\alpha}) \log\left(\frac{\mathrm{q}(\mathbf{f}, \boldsymbol{\alpha})}{\mathrm{p}(\mathbf{f}, \boldsymbol{\alpha}|\mathbf{y})}\right) \mathrm{d}(\mathbf{f}, \boldsymbol{\alpha}) = \\ &\int_{\mathbf{f},\boldsymbol{\alpha}} \mathrm{q}(\mathbf{f}, \boldsymbol{\alpha}) \log\left(\frac{\mathrm{q}(\mathbf{f}, \boldsymbol{\alpha})}{\mathrm{p}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha})}\right) \mathrm{d}(\mathbf{f}, \boldsymbol{\alpha}) + \text{const} \end{aligned} \tag{5}$$

and utilize the mean field approximation [11] $\mathrm{q}(\mathbf{f}, \boldsymbol{\alpha}) = \mathrm{q}(\mathbf{f})\mathrm{q}(\boldsymbol{\alpha})$ where $\mathrm{q}(\boldsymbol{\alpha})$ is assumed to be degenerate. The KL divergence is always non-negative and is equal to zero if and only if $\mathrm{q}(\mathbf{f}, \boldsymbol{\alpha})$ and $\mathrm{p}(\mathbf{f}, \boldsymbol{\alpha}|\mathbf{y})$ coincide. Unfortunately, the functional form of $\mathrm{p}(\mathbf{y}|\mathbf{f})$ does not allow for the direct evaluation of the KL divergence. To alleviate this problem, we use the variational bound [11] to obtain a quadratic function on $\mathbf{f}$

$$\begin{aligned} &\log \mathrm{p}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha}) \geq \sum_{j=1}^{N} \left\{ \left(y_j - \frac{1}{2}\right) \mathbf{1}^T \mathbf{f}_{\cdot,j} - \lambda(\xi_j) \mathbf{f}_{\cdot,j}^T \mathbf{1}\mathbf{1}^T \mathbf{f}_{\cdot,j} \right. \\ &\left. + \lambda(\xi_j)\xi_j^2 + \frac{\xi_j}{2} - \log(\boldsymbol{\sigma}(-\xi_j)) \right\} - \frac{1}{2}\mathbf{f}^T \mathbf{K}^{-1} \mathbf{f} - \frac{1}{2}\log|\mathbf{K}| \\ &= \log(\mathbf{M}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\xi})) \end{aligned} \tag{6}$$

where $\lambda(\xi) = \frac{1}{2\xi}\left(\frac{1}{1+e^{-\xi}} - \frac{1}{2}\right)$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)$ is a vector of additional positive parameters to be estimated. Then, we have

$$\mathrm{KL}(\mathrm{q}(\mathbf{f})\mathrm{q}(\boldsymbol{\alpha})\|\mathrm{p}(\mathbf{f}, \boldsymbol{\alpha}|\mathbf{y})) \leq \mathrm{KL}(\mathrm{q}(\mathbf{f})\mathrm{q}(\boldsymbol{\alpha})\|\mathbf{M}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\xi})),$$

and now we minimize $\mathrm{KL}(\mathrm{q}(\mathbf{f})\mathrm{q}(\boldsymbol{\alpha})\|\mathbf{M}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\xi}))$ on $\mathrm{q}(\mathbf{f})$, $\boldsymbol{\alpha}$, and $\boldsymbol{\xi}$.

It can be shown in [11] that the optimal posterior distribution approximation is

$$\mathrm{q}(\mathbf{f}) = \mathcal{N}(<\mathbf{f}>, \boldsymbol{\Sigma}) \tag{7}$$

where $\boldsymbol{\Sigma} = (\mathbf{K}^{-1} + \mathbf{W})^{-1}$ and $<\mathbf{f}> = \boldsymbol{\Sigma}\mathbf{v}$ with $\mathbf{v} = \mathbf{1} \otimes (\mathbf{y} - \frac{1}{2}\mathbf{1})$, $\mathbf{W} = 2(\mathbf{1}\mathbf{1}^T \otimes \boldsymbol{\Lambda})$, $\boldsymbol{\Lambda} = \text{diag}[\lambda(\xi_1), \ldots \lambda(\xi_N)]$, and $\otimes$ denotes the Kronecker product.

To calculate $\mathrm{q}(\mathbf{f})$, $\mathbf{K}^{-1}$ is required; however this matrix may be singular. This may happen when some sensors give correlated measurements. To tackle this problem, following [1], we use the Woodbury identity and calculate $\boldsymbol{\Sigma}$ by

$$\boldsymbol{\Sigma} = \mathbf{K} - \mathbf{K}\mathbf{W}^{1/2}(\mathbf{I} + \mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2})^{-1}\mathbf{W}^{1/2}\mathbf{K}. \tag{8}$$

To estimate $\boldsymbol{\xi}$, we maximize $<\mathbf{M}(\mathbf{y}, \mathbf{f}, \boldsymbol{\xi})>_{\mathrm{q}(\mathbf{f})}$. Taking derivatives with respect to ξ_j and equating them to zero, we obtain

$$\xi_j = \sqrt{\mathbf{1}^T \left(<\mathbf{f}_{\cdot,j}><\mathbf{f}_{\cdot,j}>^T + \Sigma_j\right)\mathbf{1}} \tag{9}$$

where $\boldsymbol{\Sigma}_j$ is obtained from $\boldsymbol{\Sigma}$ by removing the rows and columns which do not correspond to the components of $\mathbf{f}_{\cdot,j}$.

The parameters in $\boldsymbol{\alpha}$ are estimated from a lower bound of the marginal distribution of $\mathbf{y}$.

The right hand side of Eq. (6) can be written as:

$$\begin{aligned} &\log(\mathbf{M}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\xi})) = \text{const} - \frac{1}{2}\mathbf{f}^T \mathbf{K}^{-1} \mathbf{f} - \frac{1}{2}\log|\mathbf{K}| \\ &- \sum_{j=1}^{N} \frac{1}{\lambda(\xi_j)} \left\{ \frac{1}{2}\left(y_j - \frac{1}{2}\right) - \lambda(\xi_j)\mathbf{1}^T \mathbf{f}_{\cdot,j} \right\}^2 \end{aligned} \tag{10}$$

which corresponds to the log of a Gaussian distribution. We have:

$$\mathbf{z} = \sum_{i=1}^{P} \mathbf{f}_{i,\cdot} + \boldsymbol{\epsilon}, \tag{11}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 1/2\boldsymbol{\Lambda}^{-1})$ and $\mathbf{z} = \frac{\boldsymbol{\Lambda}^{-1}}{2}(\mathbf{y} - \frac{1}{2})$.

Integrating the above equation on $\mathbf{f}$ and $\boldsymbol{\epsilon}$, the marginal distribution of $\mathbf{z}$, $\mathrm{p}(\mathbf{z}|\boldsymbol{\alpha})$ is

$$\mathrm{p}(\mathbf{z}|\boldsymbol{\alpha}) = \mathcal{N}\left(\mathbf{z}|\mathbf{0}, \sum_{i=1}^{P} \alpha_i \mathbf{K}_i + \frac{\boldsymbol{\Lambda}^{-1}}{2}\right) \tag{12}$$

To estimate α_i, we differentiate $-2\log(\mathrm{p}(\mathbf{z}|\boldsymbol{\alpha}))$ with respect to α_i and utilize gradient descent to estimate this variance parameter. Alternatively, we can use the fixed point iterative procedure:

$$\alpha_i = \alpha_{i,old} \frac{\mathbf{z}^T \mathbf{R}\mathbf{K}_i\mathbf{R}\mathbf{z}}{\mathrm{Tr}\left[\mathbf{R}\mathbf{K}_i\right]}. \tag{13}$$

where $\mathbf{R} = \left(\sum_{i=1}^{P} \alpha_{i,old}\mathbf{K}_i + \frac{\boldsymbol{\Lambda}^{-1}}{2}\right)^{-1}$ which has worked well in all the experiments.

The proposed algorithm is summarized in Algorithm 1.

Algorithm 1

Require: $\mathbf{X}$, $\mathbf{y}$, $\boldsymbol{\alpha}^0 = (1, 1, \ldots, 1)^T$, and $\xi_j^0 = 1, \forall j = 1, \ldots, N$.

1: Calculate $\mathbf{K}$.
2: **repeat**
3: Calculate $\mathrm{q}^{(n+1)}(\mathbf{f})$ using Eq. (7).
4: Calculate $\boldsymbol{\xi}^{(n+1)}$ using Eq. (9).
5: **repeat**
6: Calculate $\alpha_i, i = 1, \ldots, P$ using Eq. (13).
7: **until** convergence ($\boldsymbol{\alpha}^{n+1}$ is obtained).
8: **until** convergence

Notice that in order to deal with missing data in the training phase the $j-$th row and column of $\mathbf{K}_i$ are set equal to zero if $\mathbf{x}_{ij}$ is not available.

4. CLASSIFICATION RULE

Given a new sample $\mathbf{x}^* = [\mathbf{x}_1^{*T}, \ldots \mathbf{x}_P^{*T}]^T$ and the corresponding latent variable $\mathbf{f}^* = (f_1^*, \ldots, f_P^*)^T$ the classification rule is based on the distribution

$$\mathrm{p}(\mathbf{f}^*|\mathbf{y}) = \int_{\mathbf{f}} \mathrm{p}(\mathbf{f}^*|\mathbf{f})\mathrm{p}(\mathbf{f}|\mathbf{y})\mathrm{d}\mathbf{f} \approx \int_{\mathbf{f}} \mathrm{p}(\mathbf{f}^*|\mathbf{f})\mathrm{q}(\mathbf{f})\mathrm{d}\mathbf{f} \tag{14}$$

The joint distribution $p(\mathbf{f}, \mathbf{f}^*)$ is Gaussian with zero mean and covariance matrix

$$\begin{bmatrix} \mathbf{K} & \mathbf{H} \\ \mathbf{H}^T & \mathbf{C} \end{bmatrix} \tag{15}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{h}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{h}_P \end{bmatrix}, \ \mathbf{C} = \mathrm{diag}[c_1, c_2, \ldots, c_P]$$

with

$$\mathbf{h}_i = (\alpha_i k_i(\mathbf{x}_{i1}, \mathbf{x}_i^*), \ldots, \alpha_i k_i(\mathbf{x}_{iN}, \mathbf{x}_i^*))^T \tag{16}$$

$$c_i = \alpha_i k_i(\mathbf{x}_i^*, \mathbf{x}_i^*). \tag{17}$$

Then, we have

$$\mathrm{p}(\mathbf{f}^*|\mathbf{f}) = \mathcal{N}(\mathbf{f}^*|\mathbf{H}^T\mathbf{K}^{-1}\mathbf{f}, \mathbf{C} - \mathbf{H}^T\mathbf{K}^{-1}\mathbf{H}) \tag{18}$$

which finally produces in Eq. (14)

$$\mathrm{p}(\mathbf{f}^*|\mathbf{y}) \approx \mathcal{N}(\mathbf{f}^*|\mathbf{H}^T\mathbf{K}^{-1}<\mathbf{f}>, \mathbf{C} - \mathbf{H}^T\mathbf{W}^{1/2}(\mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2} + \mathbf{I})^{-1}\mathbf{W}^{1/2}\mathbf{H}) \tag{19}$$

Since the decision boundary corresponding to $\mathbf{1}^T<\mathbf{f}^*|\mathbf{y}> = 0$ is of interest, it is only needed to consider the mean and the effect of the variance can be ignored. Therefore, the classification can be written as

$$y^* = \begin{cases} 1 & \text{if} \quad \mathbf{1}^T\mathbf{H}^T\mathbf{K}^{-1}<\mathbf{f}> \geq 0 \\ 0 & \text{if} \quad \mathbf{1}^T\mathbf{H}^T\mathbf{K}^{-1}<\mathbf{f}> < 0 \end{cases} \tag{20}$$

Notice that observing the value of $<\mathbf{f}>$ and calculating $\boldsymbol{\Sigma}$ using Eq. (8), $\mathbf{1}^T\mathbf{H}^T\mathbf{K}^{-1}<\mathbf{f}>$ can be found easily.

Notice that if at testing phase, $\mathbf{x}_i^*$ is not observed, we set $\mathbf{h}_i$ to a zero column vector and c_i to zero in Eqs. (16) and (17), respectively.

5. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed algorithm using synthetic and real datasets in different scenarios.

Fig. 1 displays the synthetic dataset used in the first experiment. This dataset is called Two-Moon and was introduced by Zhou et al. [12]. The top and bottom half moons correspond to two different classes, and they cannot be linearly separated. This is a typical example where kernel based classifiers achieve good performance. In our experiments, we use a Gaussian kernel with $\sigma^2 = 0.1$. Since Two-Moon is a 2-dimensional example, we assume that we have 2 sensors: sensor 1 measures the horizontal component of each sample while sensor 2 measures the vertical one. We start by studying the noise-free and no-missing data case. 30 samples (15 from each class) are randomly selected for training and 200 samples (100 for each class) are used for testing. To obtain unbiased results, the experiment is repeated 10 times. Data provided by each sensor are also classified independently using a GP classifier. The Overall Accuracy (OA) obtained in each realization is shown in the second and third columns of Table 1. The mean OAs for sensors 1 and 2 are 77.2% and 87.3%, respectively. Therefore, we can conclude that the information provided by sensor 2 is more discriminative for classifying the samples. The fourth column of Table 1 shows the OA obtained using the proposed method. Notice that the proposed method combines information coming from both sensors and obtains a better OA in all cases. The fifth and sixth columns show the $\boldsymbol{\alpha}$ estimated by the proposed method in 10 realizations. We observe that the α_2 values are higher than the α_1 values. It means that the proposed method detects that sensor 2 has more discriminative information than sensor 1 and is more accurate, as should be clear from Fig. 1.

Table 1. Classification accuracies for the Two-Moon Dataset in the noise-free, no-missing data case, and the estimated values of $\boldsymbol{\alpha}$ parameters.

Real.	Sensor 1	Sensor 2	Prop. Method	α_1	α_2
1	76.50	85.50	97.50	0.712	5.011
2	73.50	91.00	99.50	2.393	4.422
3	85.50	88.50	96.00	1.362	5.040
4	77.00	88.50	88.50	0.000	5.213
5	79.50	83.50	83.50	0.000	5.864
6	77.50	88.00	98.00	1.183	5.210
7	77.00	84.50	91.00	0.815	5.385
8	76.00	87.50	87.50	0.290	5.253
9	73.50	86.50	99.50	2.006	4.568
10	76.00	89.50	89.50	0.129	5.291
Mean	77.20	87.30	93.05	-	-

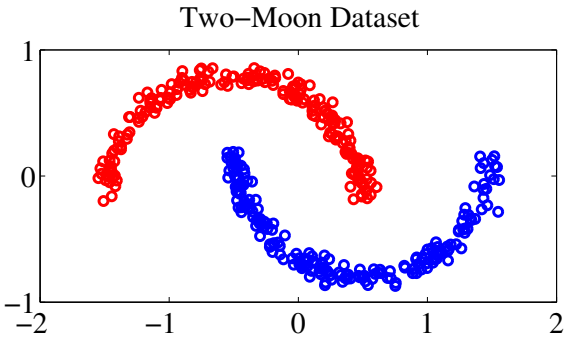

Fig. 1. Two-Moon dataset.

We now study the case in which noise is added to the samples. In this experiment, we use the same data of the ninth realization of the previous experiment (Table 1). Gaussian noise of respective variances $\sigma_1^2, \sigma_2^2 \in \{0.01, 0.04, 0.09\}$ is added to the first and second sensors independently. The experiment is repeated 10 times with each combination of noise realizations and the mean OAs are reported in Table 2. The first and second columns show the variances of the noise added in each case. The third and fourth columns show the mean OAs obtained by a GP classifier trained using the information provided by sensors 1 and 2, respectively. The fifth column shows the mean OAs obtained by the proposed method, in which the information provided by each sensor is fused to achieve a better performance. The last two columns show the values of estimated $\boldsymbol{\alpha}$ in one of the experiment realizations. Notice that for each sensor the higher the noise variance the lower the OA in the corresponding sensor.

To analyze the missing data case, $10\%, 30\%$ and 50% of samples from each sensor are randomly selected as missing samples. In this experiment, we use the same data of the ninth realization of Table 1. To obtain unbiased results, we

Table 2. Mean OAs of the Two-Moon Dataset with noisy data, and estimated $\boldsymbol{\alpha}$ parameters.

σ_1^2	σ_2^2	Sensor 1	Sensor 2	Prop. Method	α_1	α_2
0.01	0.01	71.75	86.35	96.80	2.299	4.006
0.01	0.04	71.75	84.80	94.80	2.555	3.411
0.01	0.09	71.75	83.70	93.25	2.732	2.955
0.04	0.01	70.15	86.35	93.90	1.731	4.067
0.04	0.04	70.15	84.80	93.05	2.050	3.439
0.04	0.09	70.15	83.70	89.15	2.281	3.012
0.09	0.01	67.30	86.35	90.20	0.767	3.901
0.09	0.04	67.30	84.80	89.00	1.134	3.310
0.09	0.09	67.30	83.70	87.85	1.380	2.935

repeated this experiment with each combination of missing realizations 10 times with different missing samples. The results are reported in Table 3. The first and second columns show the percentage of missing samples in each sensor. The third and fourth columns show the mean OA obtained training a GP classifier with the information provided for each sensor. The fifth column shows the mean OAs achieved by the proposed method. The last two columns show the values of estimated $\boldsymbol{\alpha}$ in an experiment realization. Again, the proposed method fuses the information provided by the two sensors to obtain a better classification performance. We can see that for a fixed percentage of missing samples of the first sensor, the difference between α_1 and α_2 decreases when the percentage of missing samples of the second sensor increases. It means that the system detects that the information provided by the second sensor is less discriminative when more samples are missing.

Table 3. Mean accuracies of classification for Two-Moon Dataset for missing data case, and estimated $\boldsymbol{\alpha}$ parameters.

% MS	% MS	Sensor 1	Sensor 2	Prop. Method	α_1	α_2
10	10	71.15	86.25	97.35	1.936	4.147
10	30	71.15	85.70	94.95	2.089	4.100
10	50	71.15	85.20	88.90	2.602	3.242
30	10	72.35	86.25	92.80	1.090	4.072
30	30	72.35	85.70	92.85	1.378	4.112
30	50	72.35	85.20	88.85	1.708	3.189
50	10	71.85	86.25	89.10	0.543	3.910
50	30	71.85	85.70	91.20	0.782	3.952
50	50	71.85	85.20	86.95	1.188	3.141

In the final experiment, the proposed method is used to solve a real multispectral image classification problem, where the goal is to classify pixels as belonging to Urban vs. Non-Urban classes. In this experiment we use a satellite image of Rome (Italy) captured in 1995. The image is composed of 7 bands obtained by Landsat TM sensor, 2 SAR backscattering intensities, the SAR interferometric coherence and a spatially filtered version of the coherence which is specially designed

to increase the urban areas discrimination [13].

In this experiment, we assume that we have 11 sensors (one for each band). The training set has 100 samples (50 from each class), and the test set 1000 samples (500 from each class). To obtain unbiased results, the experiment is repeated 10 times with different training sets. The proposed method achieved 96.25% mean OA.

Table 4. Mean classification accuracy of each 11 sensors and estimated $\boldsymbol{\alpha}$ parameters by the proposed method.

Sensor	1	2	3	4	5	6	7	8	9	10	11
OA %	65.92	66.24	81.58	85.89	79.7	82.25	83.05	59.05	77.66	79.23	91.09
$\boldsymbol{\alpha}$	0	0	30.99	34.31	0	0	0	0	0	0	133.94

The second row in table 4 shows each individual sensor mean OA using a GP classifier. The third row reports the estimated $\boldsymbol{\alpha}$ obtained by the proposed method. These parameters can be interpreted as a confidence measure on the information each sensor uses in the classification task. Notice that sensors with individual low OA obtained very small values $\alpha_i \approx 0$ in our fusion procedure, while the non-zeros α values correspond to the more accurate sensors (higher individual OAs).

A simple fusion procedure is to stack all the measures provided by all the sensors of each sample and use the stacked information matrix to build a GP based classifier. We refer to this method as STACK+GP. Notice that this method is not designed to work for sensors with missing samples. In these cases, the incomplete sample (with a missing component) must be discarded, and a large amount of information is not used in the classification task. In Table 5, we compare the proposed method with STACK+GP in the cases of 0%, 40% and 80% of missing samples. Notice that the proposed method is more robust than STACK+GP. The proposed method loses less than 0.5% of OA between the extreme cases while STACK+GP loses more than 3%.

Table 5. Mean OA of STACK+GP and proposed method with different percentages of missing samples.

Missing Samples	0%	40%	80%
STACK+GP	96.71	95.94	93.57
Proposed Method	96.31	95.77	95.88

6. CONCLUSION

In this paper, a new approach was presented using Gaussian processes in order to fuse the information of different sensors and then classify them into two groups. We showed that this method can be used in situations in which there are noisy data or even some samples have missing components. In this algorithm, based on the reliability of each sensor its weight is estimated. Therefore, under different settings, sensors may have different weights reliability. As it was confirmed by our experimental results, the more accurate sensor has a larger estimated weight. In order to evaluate the classification accuracy of the algorithm, we applied it to the Two-Moon synthetic data and showed that by using our fusion algorithm, the classification accuracy increases. Also, the algorithm was applied to a multispectral image and the weights of the sensors were reported. In this experiment, the classification accuracy increases when the fusion algorithm is applied to this binary classification problem.

7. REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[2] T. Cohn and L. Specia, "Modelling annotator bias with multi-task Gaussian processes: An application to machine translation quality estimation," in *Proc. of the 51st Annual Meeting for Computational Linguistics*, 2013, pp. 32–42.

[3] D. B. Dunson and E. B. Fox, "Multiresolution Gaussian processes," in *Advances in Neural Information Processing Systems*, 2012, pp. 737–745.

[4] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, pp. 20110550, 2013.

[5] D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive Gaussian processes," in *Advances in neural information processing systems*, 2011, pp. 226–234.

[6] M. P. Gerardo-Castro, T. Peynot, F. Ramos, and R. Fitch, "Robust multiple-sensing-modality data fusion using Gaussian process implicit surfaces," in *IEEE 17th Int. Conf. on Information Fusion*, 2014, pp. 1–8.

[7] M. Girolami, "Bayesian data fusion with Gaussian process priors: An application to protein fold recognition," in *Workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology*, 2006.

[8] A. Kapoor, H. Ahn, and R. W. Picard, "Mixture of Gaussian processes for combining multiple modalities," in *Multiple Classifier Systems*, Nikunj C. Oza, Robi Polikar, Josef Kittler, and Fabio Roli, Eds., number 3541 in Lecture Notes in Computer Science, pp. 86–96. Springer Berlin Heidelberg, Jan. 2005.

[9] F. Rodrigues, F. Pereira, and B. Ribeiro, "Gaussian process classification and active learning with multiple annotators," in *Proc. of the 31st Int. Conf. on Machine Learning*, 2014, pp. 433–441.

[10] A. R. Groves, C. F. Beckmann, S. M. Smith, and M. W. Woolrich, "Linked independent component analysis for multimodal data fusion," *Neuroimage*, vol. 54, no. 3, pp. 2198–2217, 2011.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, 2006.

[12] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, vol. 16, no. 16, pp. 321–328, 2004.

[13] L. Gómez-Chova, D. Fernández-Prieto, J. Calpe-Maravilla, E. Soria-Olivas, J. Vila-Francs, and G. Camps-Valls, "Urban monitoring using multi-temporal SAR and multi-spectral data," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 234–243, 2006.