Pattern Recognition Letters 116 (2018) 80–87

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Variational Gaussian process for multisensor classification problems ☆

Neda Rohani [a,*], Pablo Ruiz [a], Rafael Molina [b], Aggelos K. Katsaggelos [a]

[a] Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208-3118, USA
[b] Departamento de Ciencias de la Computación e I.A., Universidad de Granada, Granada 18171, Spain

ARTICLE INFO

Article history:
Received 13 February 2018
Available online 31 August 2018

Keywords:
Fusion
Gaussian process
Variational inference
Kernel
Posterior probability

ABSTRACT

This paper proposes a new model for multi-sensory data classification. To tackle this problem, probabilistic modeling and variational Bayesian inference are used. A Gaussian Process (GP) classifier is built upon the introduced modeling. Its posterior distribution is approximated using variational Bayesian inference. Finally, labels of test samples are predicted employing this classifier. Very importantly, and in contrast to alternative approaches, the proposed method does not discard samples with missing features and utilizes all available information for training. Furthermore, to take into account that the quality of the information provided by each sensor may differ (some modalities/sensors may provide more reliable/distinctive information than others), we introduce two versions of the algorithm. In the first one, the parameters modeling each sensor performance are shared while in the second one, each sensor parameters are estimated independently. Synthetic and real datasets are utilized to examine the validity of the proposed models. The results obtained for binary classification problems justify their use and confirm their superiority over existing fusion architectures.

1. Introduction

There are numerous machine learning problems where different views of a single object exist and multimodal information can be used to provide more global information on the object of interest. In such problems, different sensors (modalities) capture information and data fusion is employed to combine the information gathered by all sources, which should lead to a more accurate understanding of the environment. The more the sensors, the greater the amount of available information, and therefore, the better the performance of the system. However, fusion techniques become especially useful when the information provided by different sensors is complementary [see 13]. In these cases, the combination of the information results in an extra improvement of the performance, which would not be possible if the information of each sensor is processed separately.

Kernel based methods such as Support Vector Machines (SVM) [7] or Gaussian Processes (GP) [18] are currently two of the most utilized fusion tools. In [3], the authors propose a composite kernel machine framework for the enhanced classification of hyperspectral images.

In [1], the authors tackle an urban tree species classification problem using both AVIRIS and LIDAR data. Initially, they process the AVIRIS and LIDAR data separately, and then they apply segmentation algorithms to each data set to obtain regions of interest (ROI), and feature extraction techniques. After these three steps, the extracted features are fused (concatenated) and a canonical discriminant analysis classifier is applied to find the label of each individual pixel. Finally, each ROI is labeled by majority voting. The authors show that the classification accuracy improves when fusion techniques are used.

Multiple feature learning for hyperspectral image classification is studied in [14]. In this article, the authors use both linear and nonlinear sets of features extracted from the original spectral features. They use a combination of both types of features to cope with linear and nonlinear boundaries between different data classes. Logistic regression with a variable splitting and an augmented Lagrangian (LORSAL) algorithm is selected as the classifier for this framework.

In [9], the authors propose a Nonlinear Multiple Kernel Learning algorithm which uses spectral and spatial features for the hyperspectral images. Principal Component Analysis is performed on the original features and spatial features are extracted. Multiple kernels are nonlinearly combined. This algorithm is used for hyperspectral image classification.

☆ **Conflict of interest:** The authors confirm that no conflicts of interest associated with this article.

* Corresponding author. This work was supported in part by the National Science Foundation, award INSPIRE 15-47880, the Spanish Ministry of Economy and Competitiveness through the project DPI2016-77869-C2-2-R and the Visiting Scholar program at the University of Granada.

E-mail address: nedarohani@u.northwestern.edu (N. Rohani).

https://doi.org/10.1016/j.patrec.2018.08.035
0167-8655/

During the last few years, Deep Learning (DL) has been shown to be a powerful tool for solving fusion problems (see [17] for an extensive survey). For instance, in [11], the authors use Convolutional Neural Networks (CNNs) for fault diagnosis on a planetary gearbox. The main problem of using DL for fusion is that most of the proposed methods in the literature cannot deal with missing samples. Only generative methods, such as [5] or [10] can simulate the missing modality and use it for classification.

Regarding the model we introduce in this paper, the most similar works in the literature are [12,19] and [6]. In [12], the authors introduce one GP for modeling each sensor. The data fusion is performed in the likelihood function which is a mixture of cumulative distribution functions of a standard normal distribution. Expectation Propagation (EP) inference is used to approximate the posterior distribution of the unknowns. However, this formulation requires an extra step for estimating the weights of each sensor for classifying a new sample. Although the proposed model shares the same structure, we propose a new likelihood function and Variational Inference which allow a joint estimation of all unknowns of the model. In [19], the authors also consider one GP for modeling each sensor. For data fusion, the authors introduce a consensus function. In Section 5, we show that this consensus function is very sensitive to noisy sensors, which can lead to poor performances in some cases. In [6], the authors introduce one GP for two sensors, whose prior covariance matrix is a sum of a linear and squared exponential kernels [see 18], that is, one kernel for each sensor. We see in Section 5, that it can be formulated as a particular case of the proposed method; however the formulation proposed by the authors in [6] does not allow to deal with missing samples.

The rest of the paper is organized as follows. First, we introduce a Bayesian modeling of the fusion for classification in Section 2. Variational inference is used to derive the training algorithm in Section 3. In Section 4, we introduce the classification rule. In Section 5, we discuss the relationship of the proposed model with early and late fusion as well as the state-of-the-art method of Bayesian Co-Training. Experimental results are presented in Section 6, and Section 7 concludes the paper.

2. Bayesian modeling

The main goal in this work is to solve a classification problem where data are acquired by P different sensors. The feature training set is defined by the following matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \dots & \mathbf{x}_{1P} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \dots & \mathbf{x}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{N1} & \mathbf{x}_{N2} & \dots & \mathbf{x}_{NP} \end{bmatrix} \in \mathbb{R}^{N\times(D_1+D_2+\dots+D_P)}, \tag{1}$$

where $\mathbf{x}_{ji} \in \mathbb{R}^{1\times D_i}$ represents the jth training feature vector with dimension D_i, acquired by the ith sensor. The corresponding training labels associated to each row of $\mathbf{X}$ are given by the vector $\mathbf{y} = (y_1, \dots, y_N)^T \in \{0,1\}^{N\times 1}$. Given $\mathbf{X}$, two classical fusion strategies are possible. We can build a classifier by concatenating the features observed by all sensors, *i.e.*, by using $\mathbf{x}_j = (\mathbf{x}_{j1}, \dots, \mathbf{x}_{jP})$ with associated label y_j (this is the so called *early* fusion method). Associated to each sensor and using the same $\mathbf{y}$ for all of them, we can also build P independent classifiers, these classifiers are later combined (this is the so called *late* fusion method). While both approaches have some interests, the second one makes an independence assumption which is unrealistic in many real problems while the first one does not include explicit cross-relations between sensors whose knowledge may be of interest for the problem at hand.

We now describe the approach we propose for the multi-sensor fusion problem. To relate samples and labels, we introduce a set of latent variables for each sensor, that is, $\mathbf{f}_1, \dots, \mathbf{f}_P \in \mathbb{R}^{N\times 1}$. For the ith sensor, the corresponding set of latent variables $\mathbf{f}_i$ follows a Gaussian distribution $\mathcal{N}(\mathbf{0}, \alpha_i\mathbf{K}_i + \gamma_i^2\mathbf{I})$, where α_i is the signal variance parameter, γ_i is a Gaussian noise variance parameter, and $\mathbf{K}_i \in \mathbb{R}^{N\times N}$ is a kernel matrix depending on a set of parameters $\mathbf{\Omega}_i$. The entry (n, m) of $\mathbf{K}_i(n, m)$ is calculated as $\mathbf{K}_i(n,m) = k_{\mathbf{\Omega}_i}(\mathbf{x}_{ni}, \mathbf{x}_{mi})$ where $k_{\mathbf{\Omega}_i}(\cdot,\cdot)$ is a kernel function depending on the parameters $\mathbf{\Omega}_i$. Concatenating all latent variables, we obtain the vector $\mathbf{f} = [\mathbf{f}_1^T, \dots, \mathbf{f}_P^T]^T \in \mathbb{R}^{PN\times 1}$, which follows a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K} \in \mathbb{R}^{PN\times PN}$ is a block-diagonal matrix

$$\mathbf{K} = \begin{bmatrix} \alpha_1\mathbf{K}_1 + \gamma_1^2\mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \alpha_2\mathbf{K}_2 + \gamma_2^2\mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \alpha_P\mathbf{K}_P + \gamma_P^2\mathbf{I} \end{bmatrix}. \tag{2}$$

Eventually, during the acquisition procedure, we may have sensors that do not work appropriately, which sometimes generate samples with missing entries. Most of the proposed methods in the fusion literature are not capable of dealing with this problem, and perform the training stage by discarding all samples with missing entries, furthermore they cannot make predictions for test samples with missing features. The model proposed in this work is trained using all available information and can make predictions for test data with missing features. Assuming $\mathbf{x}_{ji}$ to be a missing feature, and since $\mathbf{x}_{ji}$ corresponds to the latent variable f_{ij}, we introduce a zero degenerate prior distribution on the latent variables corresponding to the missing data point. That is, if the ith sensor missed the information of the jth sample, we set the corresponding jth row and column of the matrix $\alpha_i\mathbf{K}_i + \gamma_i^2\mathbf{I}$ to zero.

To relate the labels $\mathbf{y}$ to the latent variables $\mathbf{f}$, we introduce the following likelihood function

$$\mathrm{p}(\mathbf{y}|\mathbf{f}) = \prod_{j=1}^{N} \mathrm{p}(y_j|\mathbf{f}_{\cdot,j}) = \prod_{j=1}^{N} \boldsymbol{\sigma}\left(\mathbf{1}^T\mathbf{f}_{\cdot,j}\right)^{y_j} \boldsymbol{\sigma}\left(-\mathbf{1}^T\mathbf{f}_{\cdot,j}\right)^{1-y_j}, \tag{3}$$

where $\boldsymbol{\sigma}(\cdot)$ is the sigmoid function, $\mathbf{f}_{\cdot,j} = (f_{1j}, \dots, f_{Pj})^T$ and $\mathbf{1}^T\mathbf{f}_{\cdot,j} = \sum_{i=1}^{P} f_{ij}$. The rationale behind this model is that each sensor is capable of providing a classifier from all information it gathers independently. For a given sample $\mathbf{x}_j$, adding the GP values associated to the 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 labels.

The joint distribution can be written as:

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

where $\mathbf{\Omega} = \{\mathbf{\Omega}_1, \dots, \mathbf{\Omega}_P\}$, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_P)^T$, $\boldsymbol{\gamma} = (\gamma_1^2, \dots, \gamma_P^2)^T$, and we use improper flat priors for $\mathrm{p}(\boldsymbol{\alpha})$, $\mathrm{p}(\boldsymbol{\gamma})$ and $\mathrm{p}(\mathbf{\Omega})$.

From a Bayesian perspective, this model has an interesting interpretation. For a given sensor, the prior distribution introduces the correlations between the samples, however it considers that the sensors are not correlated a priori as is indicated in Eq. (2). The likelihood function in Eq. (3) models how to combine the information provided by all sensors, to classify a sample. The inference procedure, that we introduce in Section 3, will result in a posterior distribution approximation, which will take into account both, correlations between samples and correlations between sensors.

3. Variational inference

The posterior distribution of the unknowns given the observations is given by $\mathrm{p}(\mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{\Omega}|\mathbf{y}) = \mathrm{p}(\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{f}, \mathbf{\Omega})/\mathrm{p}(\mathbf{y})$. However, this posterior cannot be analytically calculated because the integral $\mathrm{p}(\mathbf{y}) = \int \mathrm{p}(\mathbf{y}, \mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{\Omega})\mathrm{d}(\mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{\Omega})$ is not tractable.

Variational Bayesian Inference (VBI) approximates the posterior distribution by minimizing the Kulback–Leibler (KL) divergence

$$\mathrm{KL}(\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})||\mathrm{p}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}|\mathbf{y})) = \int \mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}) \log\left(\frac{\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})}{\mathrm{p}(\mathbf{y},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})}\right)\mathrm{d}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}) + \text{const.} \tag{5}$$

The KL divergence is always non-negative and is equal to zero if and only if $\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ and $\mathrm{p}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}|\mathbf{y})$ coincide. Unfortunately, the functional form of $\mathrm{p}(\mathbf{y}|\mathbf{f})$ does not allow the direct evaluation of the KL divergence. To alleviate this problem, we use the lower bound $\sigma(f) \geq \sigma(\xi)\exp\{(f-\xi)/2 - \lambda(\xi)(f^2-\xi^2)\}$ [see 2] which produces the following lower bound for the joint distribution

$$\mathrm{p}(\mathbf{y},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}) \geq \mathbf{M}(\mathbf{y},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega},\boldsymbol{\xi}) \propto \mathrm{p}(\mathbf{f}|\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}) \prod_{j=1}^{N}\sigma(\xi_j)\exp\left\{\left(y_j - \frac{1}{2}\right)\mathbf{1}^T\mathbf{f}_{\cdot,j} - \lambda(\xi_j)\mathbf{f}_{\cdot,j}^T\mathbf{1}\mathbf{1}^T\mathbf{f}_{\cdot,j} + \lambda(\xi_j)\xi_j^2 - \frac{\xi_j}{2}\right\} \tag{6}$$

where $\lambda(\xi) = \frac{1}{2\xi}\left(\frac{1}{1+e^{-\xi}} - \frac{1}{2}\right)$ and $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_N)^T$ is a vector of additional positive parameters to be estimated. Using the variational bound in Eq. (6), the KL divergence in Eq. (5) is upper bounded by $\mathrm{KL}(\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})\|\mathbf{M}(\mathbf{y},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega},\boldsymbol{\xi}))$, then we minimize this functional with respect to $\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ and $\boldsymbol{\xi}$, to push KL divergence in Eq. (5) to be minimum.

Additional assumptions on $\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ are imposed in order to find the solution for this minimization problem. The mean field theory [see 16] considers the following factorization for the posterior approximation $\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}) = \mathrm{q}(\mathbf{f})\mathrm{q}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ where $\mathrm{q}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ is restricted to be a degenerate distribution. The joint posterior approximation $\mathrm{q}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ can then be sequentially estimated by alternating between the estimations of $\mathrm{q}(\mathbf{f})$ and $\mathrm{q}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$.

Let $\hat{\mathrm{q}}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ and $\hat{\boldsymbol{\xi}}$ be the current estimations of $\mathrm{q}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ and $\boldsymbol{\xi}$, respectively. Then, the estimation of $\mathrm{q}(\mathbf{f})$ is given as

$$\log\hat{\mathrm{q}}(\mathbf{f}) = \mathbb{E}_{\hat{\mathrm{q}}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})}\left[\log\mathbf{M}(\mathbf{y},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega},\hat{\boldsymbol{\xi}})\right] + \text{const}, \tag{7}$$

which is a quadratic function of $\mathbf{f}$. That means $\hat{\mathrm{q}}(\mathbf{f})$ is a Gaussian distribution whose mean vector and covariance matrix can be calculated respectively by taking the first and the second derivatives of Eq. (7) with respect to $\mathbf{f}$, that leads to $\hat{\mathrm{q}}(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\bar{\mathbf{f}},\boldsymbol{\Sigma})$ where

$$\bar{\mathbf{f}} = \boldsymbol{\Sigma}\left(\mathbf{1}\otimes\left(\mathbf{y} - \frac{1}{2}\mathbf{1}\right)\right) \quad \text{and} \quad \boldsymbol{\Sigma} = (\hat{\mathbf{K}}^{-1} + \hat{\mathbf{W}})^{-1}, \tag{8}$$

with $\hat{\mathbf{W}} = 2(\mathbf{1}\mathbf{1}^T \otimes \hat{\boldsymbol{\Lambda}})$, $\hat{\boldsymbol{\Lambda}} = \mathrm{diag}[\lambda(\hat{\xi}_1),\ldots,\lambda(\hat{\xi}_N)]$ and $\otimes$ denotes the Kronecker product.

Given $\hat{\boldsymbol{\xi}}$, the approximated likelihood function $\mathbf{q}(\mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ [see 18] can be calculated by integrating $\mathbf{M}(\mathbf{y},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega},\hat{\boldsymbol{\xi}})$ on $\mathbf{f}$ resulting in

$$\mathbf{q}(\mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}) = \mathcal{N}\left(\mathbf{y}|\frac{1}{2}\mathbf{1}, 2\hat{\boldsymbol{\Lambda}} + 4\sum_{i=1}^{P}\hat{\boldsymbol{\Lambda}}\left(\alpha_i\mathbf{K}_i + \gamma_i^2\mathbf{I}\right)\hat{\boldsymbol{\Lambda}}\right), \tag{9}$$

which is used to calculate the point where $\hat{\mathrm{q}}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ degenerates as

$$\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\gamma}},\hat{\boldsymbol{\Omega}} = \arg\max_{\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}}\mathrm{q}(\mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}). \tag{10}$$

To estimate the variational parameters $\boldsymbol{\xi}$, we maximize $\mathbb{E}_{\hat{\mathrm{q}}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})}\left[\log\mathbf{M}(\mathbf{y},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\Omega,\boldsymbol{\xi})\right]$. Taking derivatives with respect to ξ_j and equating to zero, we obtain

$$\xi_j = \sqrt{\mathbf{1}^T\left(\bar{\mathbf{f}}_{\cdot,j}\bar{\mathbf{f}}_{\cdot,j}^T + \boldsymbol{\Sigma}_j\right)\mathbf{1}}, \tag{11}$$

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

The inference procedure is summarized in Algorithm 1.

Algorithm 1 Intermediate Fusion Training.

Require: $\mathbf{X}$, $\mathbf{y}$, initials $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\gamma}}$, $\hat{\boldsymbol{\Omega}}$ and $\hat{\xi}_j = 1, \forall j = 1,\ldots,N$.
1: **repeat**
2: Update $\hat{\mathrm{q}}(\mathbf{f})$ using eq.(8).
3: Update $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\gamma}}$, $\hat{\boldsymbol{\Omega}}$ by solving the problem in eq.(10).
4: Update $\hat{\boldsymbol{\xi}}$ using eq.(11).
5: **until** convergence

4. Classification rule

Given a new sample $\mathbf{x}_* = [\mathbf{x}_{*1},\ldots,\mathbf{x}_{*P}]$, the classification rule is based on the posterior probability y_*, which can be written as

$$\mathrm{p}(y_*|\mathbf{y}) = \int \mathrm{p}(y_*|\mathbf{f}_{\cdot,*})\mathrm{p}(\mathbf{f}_{\cdot,*}|\mathbf{f})\mathrm{p}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}|\mathbf{y})\mathrm{d}(\mathbf{f}_{\cdot,*},\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}), \tag{12}$$

where $\mathbf{f}_{\cdot,*} = (f_{1*},\ldots,f_{P*})^T$.

The probability $\mathrm{p}(y_*|\mathbf{f}_{\cdot,*})$ is given by Eq. (3), meanwhile the posterior distribution, $\mathrm{p}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}|\mathbf{y})$, can be approximated by $\hat{\mathrm{q}}(\mathbf{f},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega})$ obtained by Algorithm 1 at convergence.

The vector $(\mathbf{f}_{\cdot,*},\mathbf{f}^T)^T$ follows a Gaussian distribution

$$\begin{pmatrix}\mathbf{f}_{\cdot,*}\\ \mathbf{f}\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \begin{bmatrix}\mathbf{C} & \mathbf{H}^T\\ \mathbf{H} & \mathbf{K}\end{bmatrix}\right) \tag{13}$$

where

$$\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{0} & \ldots & \mathbf{0}\\ \mathbf{0} & \mathbf{h}_2 & \ldots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{h}_P\end{bmatrix}, \quad \mathbf{C} = \mathrm{diag}[c_1, c_2, \ldots, c_P], \tag{14}$$

with $\mathbf{h}_i = (\alpha_i k_{\boldsymbol{\Omega}_i}(\mathbf{x}_{1i},\mathbf{x}_{*i}),\ldots,\alpha_i k_{\boldsymbol{\Omega}_i}(\mathbf{x}_{Ni},\mathbf{x}_{*i}))^T$, and $c_i = \alpha_i k_{\boldsymbol{\Omega}_i}(\mathbf{x}_{*i},\mathbf{x}_{*i}) + \gamma_i^2$, which allows us to calculate the conditional distribution $\mathrm{p}(\mathbf{f}_{\cdot,*}|\mathbf{f}) = \mathcal{N}(\mathbf{f}_{\cdot,*}|\mathbf{H}^T\mathbf{K}^{-1}\mathbf{f}, \mathbf{C} - \mathbf{H}^T\mathbf{K}^{-1}\mathbf{H})$.

By substituting the conditional distribution $\mathrm{p}(\mathbf{f}_{\cdot,*}|\mathbf{f})$ in Eq. (12) we obtain

$$\mathrm{p}(y_* = 1|\mathbf{y}) \approx \int \sigma\left(\mathbf{1}^T\mathbf{f}_{\cdot,*}\right)\mathcal{N}(\mathbf{f}_{\cdot,*}|\mathbf{m}(\mathbf{x}_*),\mathbf{S}(\mathbf{x}_*))\mathrm{d}\mathbf{f}_{\cdot,*}, \tag{15}$$

where $\mathbf{m}(\mathbf{x}_*) = \hat{\mathbf{H}}^T\hat{\mathbf{K}}^{-1}\bar{\mathbf{f}}$ and $\mathbf{S}(\mathbf{x}_*) = \hat{\mathbf{C}} - \hat{\mathbf{H}}^T(\hat{\mathbf{K}} + \hat{\mathbf{W}}^{-1})^{-1}\hat{\mathbf{H}}$. The integral in Eq. (15) is approximated as in [2] resulting in $\mathrm{p}(y_* = 1|\mathbf{y}) \approx \sigma(m(\mathbf{x}_*)\kappa(\mathbf{x}_*))$ where $m(\mathbf{x}_*) = \mathbf{1}^T\mathbf{m}(\mathbf{x}_*)$, and $\kappa(\mathbf{x}_*) = \left(1 + \frac{\pi}{8}\mathbf{1}^T\mathbf{S}(\mathbf{x}_*)\mathbf{1}\right)^{-1/2}$. Finally, $\mathbf{x}_*$ is assigned to class 1 if $\mathrm{p}(y_* = 1|\mathbf{y})$ is greater than a given threshold δ. Notice that if at testing phase, $\mathbf{x}_{*i}$ is not observed, the proposed model can still provide a prediction for the sample $\mathbf{x}_*$. To do that, we set the corresponding $\mathbf{h}_i$ and c_i in Eq. (14) equal to zero.

5. Related models

In this section, we discuss the relationship between the proposed model and alternative fusion models based on GP.

By defining $g_j = f_{1j} + \ldots + f_{Pj}$ in Eq. (3), the proposed model in Section 2 corresponds to a GP classifier [18], with latent variables $\mathbf{g} = (g_1,\ldots,g_N)^T$ and prior distribution

$$\mathrm{p}(\mathbf{g}|\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\Omega}) = \mathcal{N}\left(\mathbf{g}|\mathbf{0},\tilde{\mathbf{K}} = \sum_{i=1}^{P}\alpha_i\mathbf{K}_i + \gamma_i^2\mathbf{I}\right). \tag{16}$$

As we will see in the experimental section, the formulation introduced in Section 2 allows us to understand how our model learns the correlation between different sensors, as well as, an intuitive modeling for the missing samples case. However, we can

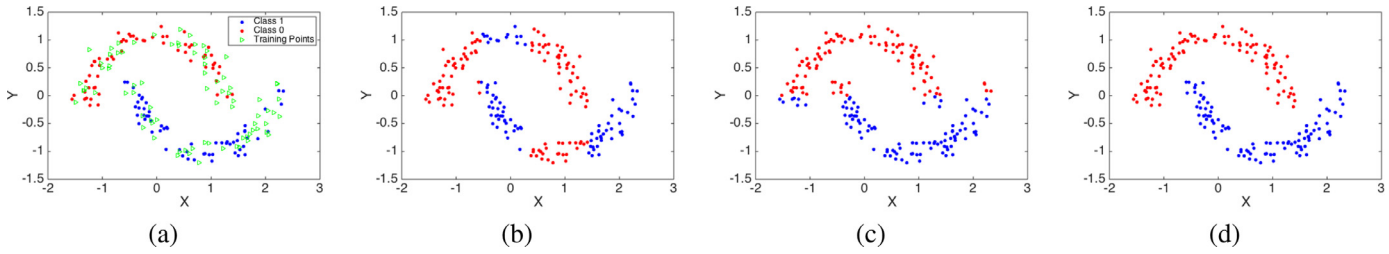

Fig. 1. (a) Original toy data set1, (b) Classification result by using *Sensor 1*, (c) Classification results by using *Sensor 2*, (d) Classification results by using *Inter 1* fusion method.

use Eq. (16) to understand how the proposed model is related to [6,19].

The prior model proposed in [6], is a particular case of the prior model in Eq. (16) when $P = 2$, $\gamma_1 = \gamma_2 = 0$, $\mathbf{K}_1$ is a linear kernel and $\mathbf{K}_2$ is a squared exponential kernel.

In [19], the authors introduce the following prior model on the latent variables

$$\mathrm{p}(\mathbf{g}_c, \mathbf{f}_1, \ldots, \mathbf{f}_P|\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) \propto \prod_{i=1}^{P} \mathcal{N}(\mathbf{f}_i|\mathbf{0}, \alpha_i\mathbf{K}_i) \exp\left\{-\frac{\|\mathbf{f}_i - \mathbf{g}_c\|^2}{2\gamma_i^2}\right\} \tag{17}$$

where $\mathbf{g}_c \in \mathbb{R}^{P\times 1}$ is a consensus function. As in our model, the factors $\mathcal{N}(\mathbf{f}_i|\mathbf{0}, \alpha_i\mathbf{K}_i)$ model the prior correlations between samples when the kernel matrices $\mathbf{K}_1, \ldots, \mathbf{K}_P$ associated to the P sensors are used. The second factor, which can be considered as a regularizer, models the relationship between the latent variables associated to the sensors as a consensus function. Notice that each vector $\mathbf{f}_i$ is forced to be similar to the latent variables $\mathbf{g}_c$. Unfortunately, when a sensor is not discriminative, the regularizer can lead to a poor behavior of $\mathbf{g}_c$ when integrating on $\mathbf{f}_1, \ldots, \mathbf{f}_P$.

By integrating in Eq. (17) on $\mathbf{f}_1, \ldots, \mathbf{f}_P$, the authors in [19] obtain the following prior model on $\mathbf{g}_c$

$$\mathrm{p}(\mathbf{g}_c|\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) = \mathcal{N}\left(\mathbf{g}_c|\mathbf{0}, \mathbf{K}_c = \left(\sum_{i=1}^{P}\left(\alpha_i\mathbf{K}_i + \gamma_i{}^2\mathbf{I}\right)^{-1}\right)^{-1}\right), \tag{18}$$

where $\mathbf{K}_c$ is called the Co-training kernel.

Using basic properties of positive definite matrices, the following relationship between the precision matrices of both prior models (the proposed one in this work in Eq. (16) and the proposed in [19] in Eq. (18) can be established

$$\mathbf{v}^T\mathbf{K}_c^{-1}\mathbf{v} \geq \mathbf{v}^T\tilde{\mathbf{K}}^{-1}\mathbf{v}, \qquad \forall\mathbf{v} \in \mathbb{R}^N. \tag{19}$$

The differences between both approaches are now clear. The Co-Training model assumes a stronger prior knowledge than our model. Our model gives more weight to the information provided by the observed labels.

6. Experimental results

In this section, the proposed approach is evaluated on both, synthetic and real data. For each sensor, we use the squared exponential kernel defined by $k_{\beta_i}(\mathbf{x}_{ji}, \mathbf{x}_{ki}) = \exp\{-\|\mathbf{x}_{ji} - \mathbf{x}_{ki}\|^2/2\beta_i^2\}$. The length scale ($\beta_i$) and signal variance parameters (α_i) are estimated for all sensors during the training step. *Inter 1* is used to denote our proposed fusion model. *Inter 2* is used for the case $\alpha_i = \alpha$, $i = 1, \ldots, P$. Both methods are compared with [3], which combines different kernels to train an SVM classifier. We name this method *CK-SVM*. Parameters are estimated following the settings proposed by Camps-Valls et al. [3], that is, the $\{\beta_i\}$ are estimated by cross-validation in the set $\beta_i \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ for $i = 1, \ldots, P$. The Bayesian co-training method proposed in [19] is also compared with our results. In order to perform a fair comparison, we consider the case when all signal variance parameters take different values for each sensor (*Co-Tr1*), and the case when all signal variance parameters take the same value (*Co-Tr2*). In the experiments, we also include early and late fusion methods. The early fusion method first stacks all features and then builds the classifier (notice that this method cannot deal with missing features). The late fusion method fuses the posterior probability provided by each of the P sensors by calculating their mean. These methods are denoted by *Early* and *Late*, respectively. We also provide the results obtained by a GP classifier applied to each sensor separately. *Sensor i* is used to denote the results obtained by the i^{th} sensor. The results of deep neural network (*DNN*) are also reported for the comparison. The network consists of three fully connected, *dense* layers. The activation functions of the first two layers are *relu* and the activation function of the last layer is a *sigmoid*. Here, the fusion is performed in the hidden layers of the deep network as is explained in [17]. The number of epochs is set to 100 and the Adam optimizer with *binary cross entropy* loss function is used.

6.1. Synthetic experiment

Fig. 1(a) displays the synthetic dataset which is used in our experiments. This dataset is called Two-moon and was introduced in [20]. The top (red) and bottom (blue) half moons correspond to two different classes, and as it can be observed from Fig. 1(a), they cannot be linearly separated.

Following the experiments presented in [12], we associate a different sensor to coordinate (dimension). "Sensor1" measures the horizontal component of each sample (X coordinate) while "Sensor2" measures the vertical one (Y coordinate). The dataset contains 200 samples, 100 from each class. For training, 40 samples from each class are randomly selected, and the remaining 120 samples are used for testing. To avoid biased results, the experiment is repeated 10 times.

In Table 1, we report the area under ROC curve (AUC), and Overall Accuracy (OA) obtained by setting the threshold value $\delta = 0.5$, for 10 realizations, as well as, the corresponding mean values reported in the last column. First and second rows report the results obtained by *Sensor 1* and *Sensor 2*, respectively. The third, fourth, fifth and sixth rows report the results obtained by *Early, Inter 1, Inter 2*, and *Late* fusion algorithms, respectively. Finally, the last three rows report the results for the state-of-the-art methods *CK-SVM, Co-Tr1* and *Co-Tr2*, respectively.

The proposed method *Inter 1* obtains 0.99 and 99.00 of mean AUC and OA, respectively, and *Inter 2* obtains 0.99 and 99.25 of mean AUC and OA, respectively. Therefore, the proposed methods can classify all samples almost perfectly. We observe that the mean OA for *Inter 2* is slightly better than *Inter 1*. Notice that, in this case, it is realistic to assume that the scale parameters are the same for the two sensors and so *Inter 2* performs slightly better than *Inter 1*. We observe the obtained results by *Sensor 1* and *Sensor 2* are much worse than *Inter 1*. This means that information

Table 1
Classification Overall Accuracies (OA) and Areas under the ROC Curves (AUC) for the compared methods.

Real.		1	2	3	4	5	6	7	8	9	10	Mean
Sensor 1	OA%	75.00	74.16	75.83	75.83	75.83	79.16	75.00	79.16	72.50	79.16	76.16
	AUC	0.82	0.84	0.86	0.86	0.85	0.87	0.85	0.88	0.84	0.86	0.85
Sensor 2	OA%	87.50	88.33	88.33	84.16	89.16	87.50	85.00	90.83	88.33	93.85	88.00
	AUC	0.96	0.96	0.96	0.96	0.97	0.95	0.95	0.97	0.96	0.98	0.96
Early	OA%	98.33	100	100	99.16	100	100	99.16	99.16	95.83	100	99.16
	AUC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99
Inter 1	OA%	98.33	98.33	100	98.33	100	100	99.16	99.16	96.66	100	99.00
	AUC	1.00	0.99	1.00	0.99	1.00	1.00	1.00	1.00	0.99	1.00	0.99
Inter 2	OA%	98.33	99.16	100	99.16	100	100	99.16	99.16	97.5	100	99.25
	AUC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99
Late	OA%	94.16	95.00	93.33	93.33	97.50	93.33	93.33	95.00	92.50	96.66	94.41
	AUC	0.96	0.98	0.98	0.97	0.99	0.98	0.96	0.98	0.97	0.99	0.98
CK-SVM	OA%	98.33	97.50	98.33	98.33	98.33	98.33	98.33	97.50	95.00	99.16	97.91
	AUC	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.00	0.99
Co-Tr1	OA%	86.66	87.50	88.33	84.16	88.33	87.50	85.00	90.00	86.66	90.83	87.50
	AUC	0.95	0.94	0.96	0.96	0.97	0.95	0.95	0.97	0.94	0.98	0.96
Co-Tr2	OA%	87.50	88.33	88.33	84.16	89.16	87.50	85.00	90.83	88.33	90.00	87.91
	AUC	0.96	0.96	0.96	0.96	0.97	0.95	0.95	0.97	0.96	0.98	0.96

Table 2
Estimated signal variance values for *Inter 1* and *Inter 2* fusion algorithms applied to Two-Moon Dataset for 10 random realizations.

Realizations		1	2	3	4	5	6	7	8	9	10
Inter 1	α_1	19.64	16.26	20.93	14.81	31.15	22.22	11.86	23.64	18.32	47.98
	α_2	114.78	144.67	132.17	139.04	155.96	137.11	120.20	149.30	100.52	223.70
Inter 2	α	46.50	47.29	54.04	47.44	63.29	58.51	45.74	56.62	38.52	95.52

provided by each sensor must be combined to obtain a good classification performance. We also observe in Table 1 that in this case *Inter 2* obtains a higher mean AUC and OA than *Early* and *Late*, which proves that performing fusion in the latent variables can lead to a better performance, than stacking the features (*Early*) or combining the classifiers outputs (*Late*). Regarding to *CK-SVM* we observe that it obtains approximately 1% of AUC and OA less than *Inter 1*. We think this happens because *CK-SVM* selects the parameters by cross-validation from a small set of values. However, the proposed parameter estimation method of *Inter 1* and *Inter 2* provides finer values of the parameters which results in a better performance. We observe that *Co-Tr2*, which uses the same value for all signal variance parameters, obtains slightly better results than *Co-Tr1* which is consistent with the results obtained by *Inter 1* and *Inter 2*. We also observe that Bayesian Co-Training obtains poor results, which are even worse than the obtained by *Sensor 2*. Notice that the information provided by *Sensor 1* is not very discriminative and the consensus function is built on contradictory information.

Fig. 1 illustrates an example of how the proposed method can combine the information provided by both sensors to obtain a better classification performance. Fig. 1(a) is the figure related to the sets used in the 5th realization in Table 1. Training samples are depicted by green triangles. Fig. 1(b) shows the classification map obtained by *Sensor 1*. As expected, many points are misclassified because of the overlap in the X dimension of points from both classes. Fig. 1(c) shows the classification map obtained by *Sensor 2*. We can observe that the number of misclassifed points is lower than in the previous case, because less points overlap in the Y direction. We can conclude the information provided by the second sensor is more discriminative for classifying samples (as we have seen in Table 1). We also observe from Fig. 1(b) and (c), that both classes cannot be perfectly separated using a linear classifier, because they overlap in the X and Y axes. Therefore, this problem only can be solved by taking into account the relationship between both sources of information. Fig. 1(d) shows the classification obtained by *Inter 1* where we observe that all points are correctly classified. Table 2 shows the estimated signal variance parameters for *Inter 1* and *Inter 2* fusion methods in 10 realizations. For *Inter 1*, we can observe that in all cases $\alpha_2 > \alpha_1$. In this case, the maximum signal variance parameter coincide with the most informative sensor. For *Inter 1*, we observe that all values for α are higher than α_1 and lower than α_2, which means that when we constrain both sensors to have the same value of signal variance, the system returns a weighted mean of the obtained values for *Inter 2*.

6.2. Radar + multispectral image classification

In this section, we investigate the use of the proposed fusion algorithms on a real dataset, where the information is provided by two sensors. We use an image from Rome (Italy) captured in 1995, the goal is to classify the pixels as belonging to Urban vs. Non-Urban classes. This image has been provided by the authors of [8] and was acquired in the context of the *Urban Expansion Monitoring Project (UrbEx)* of the *European Space Agency*.

The first sensor (ERS2 SAR) captures 2 backscattering intensities images with 35 days of difference, and returns only one intensities image ($D_1 = 1$) representing the coherence between both observations. The second sensor (Landsat TM) provides a multispectral image with $D_2 = 7$ bands. In Fig. 2(a), we plot a small part (400×200 pixels) of the coherence image captured by ERS2 SAR sensor. Fig. 2(b) shows the RGB bands captured by Landsat TM sensor for the same area. Finally, a reference land cover map provided by the Italian Institute of Statistics is also available. In Fig. 2(c), we show the region of interest corresponding to Fig. 2(a) and (b), where yellow corresponds to pixels belonging to class Urban, blue corresponds to pixels belonging to class Non-Urban and red corresponds to pixels whose class is unknown. Comparing the coherence band with the reference map, we can note a correspondence between pixels with high coherence values and pixels belonging to the class “Urban”. In the RGB image, we can also note that most of the pixels belonging to the class “No-urban” seem to have different color than pixels belonging to class “Urban”. So, both sensors

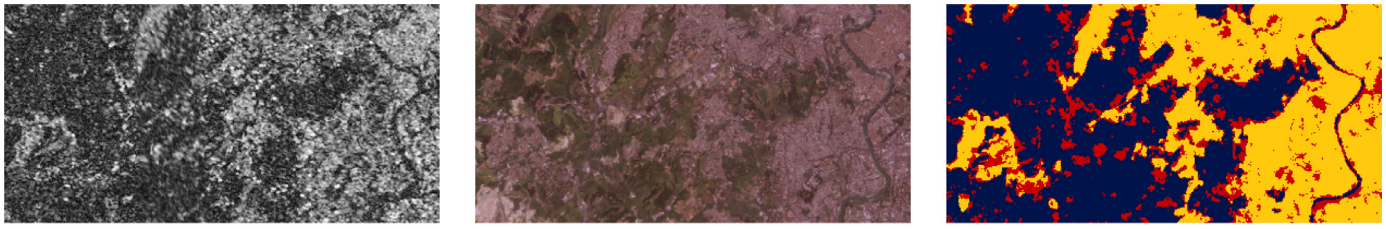

Fig. 2. Multi-spectral image classification: a) Coherence band provided by ERS2 SAR sensor, b) Original RGB image provided by Landsat TM sensor, c) Available groundtruth with Urban (YELLOW), No-urban (BLUE) and unknown pixels (RED).

Table 3
Mean classification accuracies and AUCs for probabilities p=0, p=0.4 and p=0.8 of a sensor fails acquiring a sample.

		Sensor 1	*Sensor 2*	*Early*	*Inter 1*	*Inter 2*	*Late*	*CK-SVM*	*Co-Tr1*	*Co-Tr2*	*DNN*
p=0	OA %	80.54	87.50	91.74	91.97	91.96	92.29	90.43	80.39	83.84	87.31
	AUC	0.88	0.95	0.97	0.97	0.97	0.97	0.95	0.86	0.91	0.93
p=0.4	OA %	80.53	87.42	91.26	91.84	91.87	80.13	87.49	81.63	81.43	77.18
	AUC	0.88	0.94	0.97	0.97	0.97	0.87	0.94	0.88	0.88	0.82
p=0.8	OA %	79.99	86.86	86.35	91.28	91.37	72.53	78.29	79.15	76.12	72.18
	AUC	0.87	0.94	0.95	0.97	0.97	0.80	0.84	0.86	0.82	0.78

seem to provide discriminative information for solving this classification problem. 100 pixels (50 from each class) are randomly selected for training, and 1000 pixels (500 from each class) are randomly selected for testing. To obtain unbiased results, the experiment is repeated 10 times with different training and testing sets. The first row in Table 3 shows the performance of the fusion algorithms when both sensors provide information for all training samples, i.e., the probability of malfunctioning when collecting a feature vector is $p = 0$ for both sensors. *Sensor 1* obtained 0.88 and 80.54 of mean AUC and mean OA, respectively, and *Sensor 2* obtained 0.95 and 87.50 of mean AUC and mean OA, respectively. In this case, we observe that the Landsat TM sensor has more discriminative information than the ERS2 SAR sensor. *Early* fusion obtained 0.97 and 91.74 of mean AUC and mean OA, respectively. *Inter 1* obtained 0.97 and 91.97, and *Inter 2* obtained 0.97 and 91.96 of mean AUC and mean OA, respectively. Finally, *Late* fusion obtained 0.97 and 92.29 of mean AUC and mean OA, respectively. In this case, *Late* fusion has the best overall accuracy. *Inter 1* and *Inter 2* obtained a minimum improvement over *Early* fusion. We observe that all these results are better than the obtained ones by each sensor separately. The best mean OA was obtained by *Late* fusion which was around 0.3% better than the other fusion methods. *CK-SVM* obtained 0.95 and 90.43 of mean AUC and mean OA, respectively, which means that the proposed methods were around 2% better. Bayesian Co-training algorithms have the worst performance among fusion algorithms. *Co-Tr1* obtained 0.86 and 80.39 of mean AUC and mean OA, respectively. *Co-Tr2* obtained 0.91 and 83.84 of mean AUC and mean OA, respectively. Finally, the *DNN* method obtained 0.93 and 87.31 for mean AUC and mean OA, respectively. It works better than Bayesian co-training methods and worse than the proposed method and *CK-SVM*. We believe that this is due to the small size of the training set, the low dimensionality of the space ($d = 8$) and the larger number of parameters that need to be estimated with the *DNN* algorithm.

Let us now investigate the use of sensors which do not always observe all features. For each of 10 training sets used for the no-missing samples case, we simulate the loss of information with probability $p \in \{0.4, 0.8\}$. That is, the probability of each sensor independently missing a sample is p. The results are reported in Table 3. We observe that mean AUC and mean OA drop for all methods when p increases. For $p = 0.8$, we observe that *Early* fusion obtains 2% and 5% of mean AUC and OA less than for the no missing sample case. *Late* fusion obtains 17% and 20% of mean AUC and OA less than for the no missing sample case. However, *Inter 1* and *Inter 2* obtain similar values to the no missing samples case, with a difference lower than 1% of mean AUC and mean OA. For the *CK-SVM* method, we observe that it obtains around 12% mean AUC and mean OA less than the no missing samples case. Unlike the other methods, *Inter 1* and *Inter 2* are capable of managing all information provided by both sensors. *Early* and *Late* fusion algorithms and *CK-SVM* are forced to discard all incomplete samples which leads to a poor classification performance. The missing sample case is also handled by *Co-Tr1* and *Co-Tr2*. We observe that for *Co-Tr1* the results for $p = \{0, 0.4, 0.8\}$ are very similar. However, the Co-training mean AUC and mean OA are 9% and 6% lower than the obtained by our method. It can be observed that the performance of *DNN* also degrades drastically in missing samples scenario when p increases. Here, the algorithm discards incomplete samples and is trained using less number of training samples.

We study now the behavior of the different methods when the size of the training set increases. We randomly pick training sets of sizes $100, 200, \ldots, 1000$ samples. To obtain unbiased results, each experiment is repeated 10 times. We also consider missing samples with loss probability $p \in \{0, 0.4, 0.8\}$. The averaged OA and AUC for each case are shown in Fig. 3. The x-axis represents the size of the training set and the y-axis shows the mean OA (left) or mean AUC (right). Fig. 3(a), corresponds to the perfect sensor case ($p = 0$). We observe that the proposed algorithms have the highest performance, which increases when the size of the training set also increases. The performance of the *DNN* is worse than the proposed method. We believe that this happens due to the fact that here the size of the training set is not very large and also the dimension of the dataset is low. Note that the slope of the black line is larger which confirms that the improvement observed by the *DNN* is larger when we have a larger number of training samples. With all algorithms, the performance improves with the number of training samples. In Fig. 3 (b) and (c), we plot the mean OA and mean AUC for $p = 0.4$ and $p = 0.8$, respectively. We observe that the performance of the proposed algorithms does not decrease drastically when incomplete training samples are present. However, other algorithms suffer from that, more specifically the performance of the *DNN* becomes much worse compared to the perfect sensor scenario which again verifies its dependency on the number of training samples. As expected, in all cases the perfor-

Table 4
Classification accuracies and area under the curve of different level fusion algorithms and composite kernel applied to multispectral image with perfect simulated sensors for 10 random realizations.

Real.		1	2	3	4	5	6	7	8	9	10	Mean
Sen. 1	OA%	78.55	79.60	74.95	79.15	76.70	79.00	77.20	79.85	80.40	77.65	78.30
	AUC	0.88	0.87	0.86	0.88	0.87	0.86	0.87	0.87	0.88	0.87	0.87
Sen. 2	OA%	77.30	79.75	74.75	79.30	77.80	78.35	77.15	79.95	79.50	76.65	78.05
	AUC	0.87	0.87	0.85	0.87	0.86	0.86	0.86	0.87	0.87	0.87	0.86
Sen. 3	OA%	78.00	77.75	74.70	79.10	79.15	77.65	76.60	80.05	78.90	76.55	77.84
	AUC	0.87	0.86	0.84	0.86	0.86	0.86	0.85	0.87	0.86	0.87	0.86
Sen. 4	OA%	72.95	72.50	71.30	71.85	72.20	71.60	70.65	72.50	72.65	73.05	72.12
	AUC	0.80	0.77	0.78	0.78	0.79	0.78	0.76	0.79	0.78	0.79	0.78
Early	OA%	81.75	80.40	76.65	81.35	82.05	81.75	78.65	82.85	81.80	82.60	80.98
	AUC	0.90	0.89	0.87	0.89	0.89	0.89	0.88	0.90	0.89	0.90	0.89
Inter 1	OA%	82.95	81.40	79.55	80.70	82.60	81.05	80.05	82.65	82.75	84.60	81.83
	AUC	0.90	0.89	0.89	0.89	0.89	0.89	0.88	0.91	0.90	0.91	0.89
Inter 2	OA%	83.15	81.90	79.30	81.65	82.85	81.55	80.00	82.80	83.10	83.35	81.96
	AUC	0.90	0.90	0.89	0.89	0.90	0.89	0.89	0.90	0.90	0.90	0.90
Late	OA%	81.65	81.15	75.55	79.40	80.95	78.85	78.30	81.50	80.95	78.95	79.72
	AUC	0.89	0.88	0.86	0.88	0.88	0.87	0.86	0.89	0.87	0.88	0.88
CK-SVM	OA%	79.20	79.70	75.55	76.05	75.40	78.30	75.35	79.75	79.35	77.90	77.65
	AUC	0.83	0.81	0.82	0.84	0.81	0.86	0.82	0.86	0.87	0.85	0.84
Co-Tr1	OA%	78.65	77.60	77.80	79.10	77.15	76.85	76.75	78.60	75.10	79.65	77.72
	AUC	0.85	0.86	0.87	0.87	0.87	0.86	0.85	0.87	0.85	0.87	0.86
Co-Tr2	OA%	76.55	77.85	76.90	79.40	77.15	77.00	76.85	78.95	75.10	79.00	77.47
	AUC	0.86	0.86	0.86	0.87	0.87	0.86	0.85	0.87	0.85	0.87	0.86
DNN	OA%	76.35	79.2	75.7	74.2	78.9	76.5	79.45	81.65	74.5	77.8	77.42
	AUC	0.85	0.86	0.85	0.83	0.87	0.84	0.87	0.89	0.81	0.82	0.85

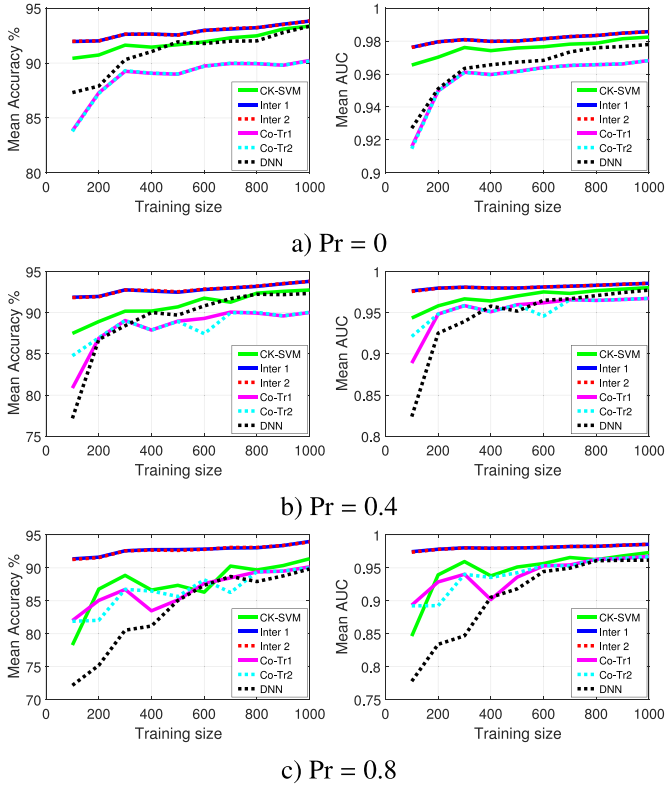

Fig. 3. Overall accuracy and area under the ROC curve vs training set size for missing case scenario with three different missing probabilities.

mance of the algorithms improves by adding more samples to the training set.

6.3. Pervasive change detection

In this experiment, the dataset was provided by the authors of [15]. The dataset is a pair of multi-spectral images captured by Quickbird satellite. Both 864×1060 images were acquired over Denver (USA) on July 17th 2002 and August 22nd 2008, respectively. The goal of this experiment is to detect pervasive changes in the images. To carry out this task, a set of 2280 labeled pixels is available, where 1140 correspond to the class $\mathcal{C}_1$ ="pervasive change", and 1140 correspond to the class $\mathcal{C}_0$ ="non-pervasive change". Quickbird multispectral images are acquired by $P = 4$ sensors capturing different wavelengths: blue, green, red and near-infrared. For each sensor, we have a $D_j = 2, j = 1, 2, 3, 4,$ feature vector where the first component corresponds to the image captured on Jul 17th 2002, and the second component corresponds to the image captured on Aug 22nd 2008. For training, 280 samples (140 from each class) are randomly selected and the remaining 2000 labeled samples (1000 from each class) are used for testing. To obtain unbiased results, the experiment is repeated 10 times. Table 4 shows the results obtained by the different algorithms. *Sensor 1* obtained 0.87 and 78.30 of mean AUC and mean OA, respectively, *Sensor 2* obtained 0.86 and 78.05 of mean AUC and mean OA, respectively, *Sensor 3* obtained 0.86 and 77.84 of mean AUC and mean OA, respectively and *Sensor 4* obtained 0.78 and 72.12 of mean AUC and mean OA, respectively. From these results, we observe that the first sensor provides the most discriminative information and the fourth sensor is the least accurate one. *Early* fusion obtained 0.89 and 80.98 of mean AUC and mean OA, respectively. *Inter 1* obtained 0.89 and 81.83, and *Inter 2* obtained 0.90 and 81.96 of mean AUC and mean OA, respectively. *Late* fusion obtained 0.88 and 79.72 of mean AUC and mean OA, respectively. In this case, *Inter 1* and *Inter 2* obtained a minimum improvement over Early and Late fusion methods. The accuracies of three fusion algorithms, *Early*, *Inter 1* and *Inter 2* are very close to each other in all runs. The last three rows of Table 4 show the results for *CK-SVM*, *Co-Tr1* and *Co-Tr2*. *CK-SVM* obtained 0.84 and 77.65 of mean AUC and mean OA, respectively. The other two performed similarly, which means that the proposed method was around 5% better than the ones compared with. Finally, *DNN* method obtained 0.85 and 77.42 of mean AUC and mean OA, respectively. It works similar to Bayesian co-training methods and worse than proposed method and *CK-SVM*. We believe this happens due to small size of training set, low dimensionality of the space ($d = 8$) and more number of parameters to be estimated in *DNN* algorithm.

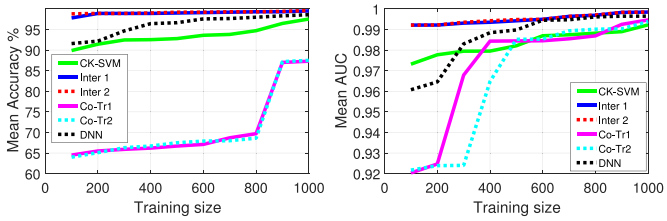

Fig. 4. Overall accuracy and Area under ROC curve vs training set size for Occupancy Detection dataset.

6.4. Occupancy detection

The dataset for this experiment was collected by the authors of [4] and is available in *UCI Machine Learning Repository*. An office room with approximate dimensions of 5.85 m $\times$ 3.50 m $\times$ 3.53 m (W $\times$ D $\times$ H) was monitored with different sensors to obtain the following variables: temperature, humidity, light and CO2 levels [4]. The goal is to detect if the room is occupied or not according to the values of the four acquired features.

To study the behavior of the different methods when the size of the training set varies, we randomly pick training sets of sizes $100, 200, \ldots, 1000$ samples. To obtain unbiased results for each number of training samples, the experiment is repeated 10 times. The testing dataset is fixed and contains 2665 samples. As we can observe in Fig. 4, the proposed algorithm outperforms the rest of the fusion algorithms. For this dataset, the Bayesian Co-training algorithm demonstrated the worst performance, whereas for the larger training sets, the performance of the *DNN* is comparable to the intermediate fusion algorithms.

7. Conclusions

In this paper, we use Gaussian Process theory to model a classification problem where information is provided by different sensors. We introduce a prior model which exploits the correlations between the samples provided by each sensor and a likelihood function which links the information provided by the sensors. Variational Bayes inference is used to approximate the posterior distribution of the model unknowns. In contrast to other methods in the literature, our proposed model can handle sensors which do not always observe all their associated features. The method is trained with the available information and provides predictions for complete as well as incomplete testing samples. We also studied the relationship with the Bayesian Co-training model and demonstrated which are the main advantages of the proposed model. In the experimental section, the proposed method is evaluated on both synthetic and real datasets, and compared with other methods in the literature. The results justified the applicability of the proposed algorithm.

References

[1] M. Alonzo, B. Bookhagen, D. Roberts, Urban tree species mapping using hyperspectral and lidar data fusion, Remote Sens. Environ. 148 (2014) 70–83.
[2] C. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag New York, Inc., NJ, USA, 2006.
[3] G. Camps-Valls, L. Gómez-Chova, J. Muñoz Marí, J. Vila-Francés, J. Calpe-Maravilla, Composite kernels for hyperspectral image classification, IEEE Geosci. Remote Sensing Lett. 3 (1) (2006).
[4] L.M. Candanedo, V. Feldheim, Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models, Energy Build. 112 (2016) 28–39.
[5] Y. Cao, S. Steffey, J. He, D. Xiao, C. Tao, P. Chen, H. Mller, Medical image retrieval: a multimodal approach, Cancer Inform. 13s3 (2014).
[6] S.-H. Chen, J.-C. Wang, W.-C. Hsieh, Y.-H. Chin, C.-W. Ho, C.-H. Wu, Speech emotion classification using multiple kernel gaussian process, in: APSIPA, 2016 Asia-Pacific, 2016, pp. 1–4.
[7] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[8] L. Gómez-Chova, D. Fernández-Prieto, J. Calpe-Maravilla, E. Soria-Olivas, J. Vila-Francés, G. Camps-Valls, Urban monitoring using multi-temporal sar and multi-spectral data., Pattern Recog. Lett. 27 (4) (2006) 234–243.
[9] Y. Gu, T. Liu, X. Jia, J. Benediktsson, J. Chanussot, Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 54 (6) (2016) 3235–3247.
[10] Y. Huang, W. Wang, L. Wang, Unconstrained multimodal multi-label learning, IEEE Trans. Multimedia 17 (11) (2015) 1923–1935.
[11] L. Jing, T. Wang, M. Zhao, P. Wang, An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox, Sensors 17 (2) (2017) 414.
[12] A. Kapoor, H. Ahn, R. Picard, Mixture of gaussian processes for combining multiple modalities, in: N. Oza, R. Polikar, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 86–96.
[13] A. Katsaggelos, S. Bahaadini, R. Molina, Audiovisual fusion: challenges and new approaches, Proc. IEEE 103 (9) (2015) 1635–1653.
[14] J. Li, X. Huang, P. Gamba, J. Bioucas-Dias, L. Zhang, J. Benediktsson, A. Plaza, Multiple feature learning for hyperspectral image classification, IEEE T. Geosci. Remote 53 (3) (2015) 1592–1606.
[15] N. Longbotham, G. Camps-Valls, A family of kernel anomaly change detectors, in: IEEE Workshop on Hyperspectral Image and Sig. Proc., 2014.
[16] G. Parisi, Statistical Field Theory, Edicin: new edition, Perseus Books, Reading, Mass, 1998.
[17] D. Ramachandram, G.W. Taylor, Deep multimodal learning: a survey on recent advances and trends, IEEE Signal Proc. Mag. 34 (6) (2017) 96–108.
[18] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2006.
[19] S. Yu, B. Krishnapuram, R. Rosales, R. Rao, Bayesian co-training, J. Mach. Learn. Res. 12 (2011) 2649–2680.
[20] D. Zhou, O. Bousquet, T. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, NIPS 16 (16) (2004) 321–328.