

LEARNING FILTERS IN GAUSSIAN PROCESS CLASSIFICATION PROBLEMS

Pablo Ruiz^{1}, Javier Mateos¹, Rafael Molina¹ and Aggelos K. Katsaggelos²*

¹ Dpto. de Ciencia de la Computación e I.A. Universidad de Granada.

² Dpt. of Electrical Engineering and Computer Science. Northwestern University.

*e-mail:mataran@decsai.ugr.es

ABSTRACT

Many real classification tasks are oriented to sequence (neighbor) labeling, that is, assigning a label to every sample of a signal while taking into account the sequentiality (or neighborhood) of the samples. This is normally approached by first filtering the data and then performing classification. In consequence, both processes are optimized separately, with no guarantee of global optimality. In this work we utilize Bayesian modeling and inference to jointly learn a classifier and estimate an optimal filterbank. Variational Bayesian inference is used to approximate the posterior distributions of all unknowns, resulting in an iterative procedure to estimate the classifier parameters and the filterbank coefficients. In the experimental section we show, using synthetic and real data, that the proposed method compares favorably with other classification/filtering approaches, without the need of parameter tuning.

Index Terms— Gaussian Process classification, filter estimation, analysis representation.

1. INTRODUCTION

Many real classification tasks assign a label to every sample of a signal (or pixel of an image) while taking into account the sequentiality (or vicinity) of the samples. This task is normally approached by first filtering the data and then performing classification. For instance, a super resolution method can be applied to a multispectral image [1] followed by a classification method on the improved multispectral image [2]; or an improved passive millimeter-wave image can be obtained [3] followed by an object detection procedure [4].

Using filtering as a pre-processing step before learning a classifier does not guarantee optimal joint performance. To solve this problem, we propose a Bayesian framework to learn a classifier, at the same time estimate an optimal filterbank to improve the classifier performance.

Let us assume that we have access to a multichannel sequential signal or multichannel sequential features extracted from the signal. We use the term “multichannel features” to refer to both concepts for simplicity. Let $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_N]$ be the matrix including these original input features, where each feature \mathbf{z}_i is of length B . Instead of performing classification directly on the features \mathbf{Z} , we would like to compute new features $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ so as to optimize the classification performance. \mathbf{Z} and \mathbf{X} can be related in two different ways. The first method, based on the analysis representation, obtains \mathbf{X} as a linear transformation of \mathbf{Z} , leading to $\mathbf{X} = \mathbf{AZ}$ where \mathbf{A} defines a linear filterbank whose coefficients must be estimated. The

analysis representation appears in many signal reconstruction problems. For instance, it can be used to improve the classification of EEG data in brain-computer interfaces [5], or to discover causality interaction in functional MRI [6].

In the second method, based on the synthesis representation, \mathbf{Z} is represented using a dictionary \mathbf{D} that has to be learnt from a set of samples, that is, $\mathbf{Z} = \mathbf{DX}$. The new features \mathbf{X} are to be used to classify the samples. The synthesis representation model is related, for instance, to the use of discriminative Gaussian Process Latent Variable Models (GPLVM) [7], where a linear discriminant prior on the latent variables is introduced and bears some connections with learning discriminative dictionaries (see, for instance, [8, 9]). In this work we use the analysis representation.

The idea of jointly optimizing a filter and a classifier dates back to the 1990s within the field of artificial neural networks. It was, for instance, used in convolutional networks [10] or to define a neural model for temporal processing [11, 12]. Recently, the same principle is used in [13] where filters are learnt jointly with a support vector machine (SVM) to perform classification.

In this work the filtering/classification tasks are formulated as a single Bayesian inference problem. Variational inference is used to learn the classifier and the optimal filterbank coefficients as well as the model parameters. The rest of this paper is organized as follows. In Section 2 Bayesian modeling is presented to use analysis representation on images. Variational Inference is performed in Section 3. The classification rule is introduced in Section 4. In Section 5 results for both synthetic and real experiments are presented and finally Section 6 concludes the paper.

2. HIERARCHICAL BAYESIAN MODELING

Let us assume that during the training phase we have access to $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_N]$, where each \mathbf{z}_i is of length B , and their corresponding labels $\mathbf{y} = [y_1, \dots, y_N]^T$ with $y_i \in \{0, 1\}$. To obtain the new features each band is filtered with a spatial filter $\mathbf{a}_i \in \mathbb{R}^{k^2}$, $i = 1, \dots, B$, producing

$$\mathbf{X} = \mathbf{AZ} = \begin{bmatrix} \mathbf{a}_1^T & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{a}_2^T & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{a}_B^T \end{bmatrix} \begin{bmatrix} \mathbf{z}_{1,1} & \mathbf{z}_{1,2} & \dots & \mathbf{z}_{1,N} \\ \mathbf{z}_{2,1} & \mathbf{z}_{2,2} & \dots & \mathbf{z}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_{B,1} & \mathbf{z}_{B,2} & \dots & \mathbf{z}_{B,N} \end{bmatrix}, \quad (1)$$

where each $\mathbf{z}_{i,j}$ is a column vector of size k^2 containing the neighborhood of the j -th sample in the i -th band. To reduce the number of coefficients in \mathbf{A} to be estimated, we only perform intraband filtering. Interband filtering is not performed because the classifier utilizes multiband information.

This work has been supported in part by the Comisión Nacional de Ciencia y Tecnología under contract TIN2010-15137, CEI BioTic at the University of Granada, and the Department of Energy grant DE-NA0000457.

To model the classification function relating each sample \mathbf{x}_i to its corresponding label y_i , we follow a two stage procedure. First, we introduce a latent variable f_i which is related to y_i by a sigmoidal function $y_i = \sigma(f_i) = 1/(1 + e^{-f_i})$. Let $\mathbf{f} = [f_1, \dots, f_N]$ be the values of the latent function at $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, then the joint likelihood factorizes to

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^N [\sigma(f_i)]^{y_i} [1 - \sigma(f_i)]^{1-y_i}. \quad (2)$$

In the second stage, to model \mathbf{f} , we define on f_i a Gaussian Process, which depends on \mathbf{X} , and so we write

$$p(\mathbf{f}|\mathbf{X}, \mu, \gamma, \sigma) = \mathcal{N}(\mathbf{f}|\mu\mathbf{1}, \mathbf{C}), \quad (3)$$

where $\mathbf{C} = \gamma\mathbf{K}_{\mathbf{X}} + \sigma\mathbf{I}$, and $\mathbf{K}_{\mathbf{X}} = (\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j))$, $i, j = 1, \dots, N$, is the kernel used. In this work linear and Gaussian kernels are considered (see [14] for details).

To model \mathbf{X} , instead of enforcing Eq. (1), we consider a weaker constraint by defining the following pseudo-observation model

$$p(\mathbf{X}|\mathbf{Z}, \mathbf{A}) \propto \exp\left(-\frac{\beta}{2}\|\mathbf{X} - \mathbf{AZ}\|_{\text{F}}^2\right), \quad (4)$$

where $\|\cdot\|_{\text{F}}$ is the Frobenius norm. When $\beta \rightarrow \infty$ we obtain the constraint $\mathbf{X} = \mathbf{AZ}$. In Sect. 3 we explain how to configure the penalty β .

With no much prior information on the filterbank coefficients, we follow the approach in [13] and use the following prior on \mathbf{A} ,

$$p(\mathbf{A}|\boldsymbol{\alpha}) = \prod_{i=1}^B p(\mathbf{a}_i|\alpha_i) = \prod_{i=1}^B \mathcal{N}(\mathbf{a}_i|\mathbf{0}, \alpha_i^{-1}\mathbf{I}_{k^2}), \quad (5)$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_B)^T$ are the precision coefficients, which are modeled using Gamma distributions, that is,

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^B p(\alpha_i) \propto \prod_{i=1}^B \alpha_i^{a_i-1} \exp(-b_i\alpha_i). \quad (6)$$

The parameters a_i and b_i are treated as deterministic whose values are set to small values (e.g., 10^{-5}) to obtain broad hyperpriors.

Finally, the joint distributions factorizes as

$$p(\mathbf{y}, \Theta) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X}, \mu, \gamma, \sigma)p(\mathbf{X}|\mathbf{A}, \mathbf{Z}, \beta)p(\mathbf{A}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}).$$

where $\Theta = \{\mathbf{f}, \mathbf{X}, \mathbf{A}, \boldsymbol{\alpha}, \mu, \gamma, \sigma\}$, and \mathbf{Z} is fixed.

3. BAYESIAN INFERENCE AND VARIATIONAL APPROXIMATION

In our Bayesian framework, unknown variables are estimated from the posterior distribution $p(\Theta|\mathbf{y}) = p(\mathbf{y}, \Theta)/p(\mathbf{y})$. However this distribution is not tractable because $p(\mathbf{y})$ can not be calculated. To alleviate this problem, variational methods are used to approximate it by a tractable distribution of the form

$$q(\Theta) = q(\mathbf{f})q(\mathbf{X})q(\mu)q(\gamma)q(\sigma) \prod_{i=1}^B q(\mathbf{a}_i)q(\alpha_i). \quad (7)$$

The variational criterion used to find $q(\Theta)$ is the minimization of the Kullback-Leibler (KL) divergence [14], given by

$$C_{\text{KL}}(q(\Theta)||p(\Theta|\mathbf{y})) = \int q(\Theta) \log \frac{q(\Theta)}{p(\mathbf{y}, \Theta)} d\Theta + \text{const} \quad (8)$$

which is always non negative and equal zero if and only if the distributions $q(\Theta)$ and $p(\Theta|\mathbf{y})$ coincide.

Due to the form of the joint likelihood defined in Eq. (2), the KL divergence cannot be evaluated. To solve this problem we bound the joint likelihood in Eq. (2), using the variational lower bound [14, 15]

$$\ln(1 + e^u) \leq \lambda(\xi)(u^2 - \xi^2) + \frac{u - \xi}{2} + \ln(1 + e^\xi), \quad (9)$$

where $\lambda(\xi) = \frac{1}{2\xi} \left(\frac{1}{1+e^{-\xi}} - \frac{1}{2} \right)$. Thus the joint likelihood is bounded as:

$$p(\mathbf{y}|\mathbf{f}) \geq \exp\left\{ \left(\mathbf{y} - \frac{1}{2}\mathbf{1}\right)^T \mathbf{f} - \mathbf{f}^T \Lambda \mathbf{f} \right\} \times \exp\left\{ \boldsymbol{\xi}^T \Lambda \boldsymbol{\xi} + \frac{1}{2}\mathbf{1}^T \boldsymbol{\xi} \right\} \prod_{i=1}^N \sigma(-\xi_i) = \mathbf{H}(\mathbf{y}, \mathbf{f}, \boldsymbol{\xi}), \quad (10)$$

where $\boldsymbol{\xi} = (\xi_1, \dots, \xi_N)^T$, and $\Lambda = \text{Diag}(\lambda(\xi_1), \dots, \lambda(\xi_N))$. The inequality in Eq. (10) leads to the following lower bound for the joint probability distribution:

$$p(\mathbf{y}, \Theta) \geq \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) = \mathbf{H}(\mathbf{y}, \mathbf{f}, \boldsymbol{\xi})p(\mathbf{f}|\mathbf{X}, \mu, \gamma, \sigma)p(\mathbf{X}|\mathbf{A}, \mathbf{Z}, \beta)p(\mathbf{A}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}). \quad (11)$$

Finally, the KL divergence in Eq. (8) is majorized by

$$C_{\text{KL}}(q(\Theta)||p(\Theta|\mathbf{y})) \leq C_{\text{KL}}(q(\Theta)||\mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi})) + \text{const}. \quad (12)$$

Although a new set of unknowns $\boldsymbol{\xi}$ has been included, now the KL divergence between $q(\Theta)$ and $\mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi})$ is mathematically tractable, and it can be used to calculate the posterior distribution $q(\Theta)$. The optimal posterior distribution approximation is the given by [14]

$$q(\theta) \propto \exp[\langle \log \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) \rangle_{q(\Theta_\theta)}], \quad (13)$$

where $\theta \in \Theta$, the set Θ_θ represents the set difference $\Theta \setminus \{\theta\}$ and the operator $\langle \cdot \rangle_{q(\Theta_\theta)}$ denotes expected value with respect to the distribution $q(\Theta_\theta)$. For simplicity we use $\langle \mathbf{u} \rangle$ to denote $\langle \mathbf{u} \rangle_{q(\mathbf{u})}$. In this paper we assume that $q(\mathbf{X})$, $q(\mu)$, $q(\gamma)$ and $q(\sigma)$ are degenerate distributions. No constraints are imposed on $q(\mathbf{f})$, $q(\mathbf{a}_i)$ and $q(\alpha_i)$.

Since $\langle \log \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) \rangle_{q(\Theta_\mathbf{f})}$ is a quadratic function on \mathbf{f} , its posterior distribution approximation is a Gaussian distribution with parameters

$$\boldsymbol{\mu}_{\mathbf{f}} = \Sigma_{\mathbf{f}} \left[\mathbf{y} - \frac{1}{2}\mathbf{1} + \mu\mathbf{C}^{-1}\mathbf{1} \right], \quad \Sigma_{\mathbf{f}} = (\mathbf{C}^{-1} + 2\Lambda)^{-1}. \quad (14)$$

The value where $q(\mu)$ is degenerate is obtained by solving

$$\hat{\mu} = \arg \min_{\mu} \langle \log \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) \rangle_{q(\Theta_\mu)}. \quad (15)$$

By differentiating $\langle \log \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) \rangle_{q(\Theta_\mu)}$ with respect to μ and equating to zero we obtain

$$\hat{\mu} = \frac{(\boldsymbol{\mu}_{\mathbf{f}})^T \mathbf{C}^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}}. \quad (16)$$

Following the same procedure for γ in Eq. (3) we obtain

$$(\boldsymbol{\mu}\mathbf{1} - \boldsymbol{\mu}_{\mathbf{f}})^T \mathbf{C}^{-1} \mathbf{K}_{\mathbf{X}} \mathbf{C}^{-1} (\boldsymbol{\mu}\mathbf{1} - \boldsymbol{\mu}_{\mathbf{f}}) - \text{Tr}[\mathbf{C}^{-1} \Sigma_{\mathbf{f}} \Lambda \mathbf{K}_{\mathbf{X}}] = 0,$$

where γ is included in \mathbf{C} . Then we use the following fixed point algorithm (see [2] for details) to update γ

$$\gamma = \frac{\gamma (\boldsymbol{\mu}\mathbf{1} - \boldsymbol{\mu}_{\mathbf{f}})^T \mathbf{C}^{-1} \mathbf{K}_{\mathbf{X}} \mathbf{C}^{-1} (\boldsymbol{\mu}\mathbf{1} - \boldsymbol{\mu}_{\mathbf{f}})}{2 \text{Tr}[\mathbf{C}^{-1} \Sigma_{\mathbf{f}} \Lambda \mathbf{K}_{\mathbf{X}}]}, \quad (17)$$

where the old value of γ is used in the right hand side to obtain an updated value in the left hand side. The same procedure is used on σ to obtain the updating rule

$$\sigma = \frac{\sigma(\boldsymbol{\mu}_f - \boldsymbol{\mu}_f)^T \mathbf{C}^{-1} \mathbf{C}^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f)}{2 \text{Tr}[\mathbf{C}^{-1} \boldsymbol{\Sigma}_f \boldsymbol{\Lambda}]}. \quad (18)$$

To estimate $\boldsymbol{\xi}$ we solve the optimization problems

$$\hat{\boldsymbol{\xi}}_i = \arg \min_{\boldsymbol{\xi}_i} \langle \log \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) \rangle_{q(\Theta)}. \quad (19)$$

Differentiating and equating to zero we obtain

$$\hat{\boldsymbol{\xi}}_i = \sqrt{\langle \boldsymbol{\mu}_f \rangle_i^2 + (\boldsymbol{\Sigma}_f)_{ii}}. \quad (20)$$

Since $\langle \log \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) \rangle_{q(\Theta_{\mathbf{a}_i})}$ is a quadratic function on \mathbf{a}_i , $q(\mathbf{a}_i)$ is a Gaussian distribution with parameters

$$\langle \mathbf{a}_i \rangle = \beta \boldsymbol{\Sigma}_i \mathbf{Z}_i (\mathbf{X}_i)^T, \quad \boldsymbol{\Sigma}_i = (\beta \mathbf{Z}_i \mathbf{Z}_i^T + \langle \alpha_i \rangle \mathbf{I}_{k_2})^{-1}, \quad (21)$$

where \mathbf{X}_i , $i = 1 \dots, B$, represent the i -th row of \mathbf{X} .

The posterior density of α_i becomes a Gamma distribution with mean

$$\langle \alpha_i \rangle = \frac{2a_i + k^2}{2b_i + \text{Tr}(\boldsymbol{\Sigma}_i + \langle \mathbf{a}_i \rangle \langle \mathbf{a}_i \rangle^T)}. \quad (22)$$

Finally, to estimate \mathbf{X} we solve

$$\hat{\mathbf{X}} = \arg \min_{\mathbf{X}} \langle \log \mathbf{M}(\mathbf{y}, \Theta, \boldsymbol{\xi}) \rangle_{q(\Theta_{\mathbf{X}})}. \quad (23)$$

For a linear kernel we have

$$\mathbf{X}^T = \left[\gamma (\mathbf{C}^{-1} \boldsymbol{\Sigma}_f 2\boldsymbol{\Lambda}) - \gamma \mathbf{C}^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f) (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f)^T \mathbf{C}^{-1} + \beta \mathbf{I}_N \right]^{-1} \beta \mathbf{Z}^T \langle \mathbf{A} \rangle^T. \quad (24)$$

For a Gaussian kernel case with a fixed scale parameter, we obtain an update rule for each component of \mathbf{X} . Thus, for the p -th component of \mathbf{x}_i we obtain

$$x_{pi} = \frac{\frac{\gamma}{s^2} \sum_{t \neq i}^N (v_i v_t - w_{it} (\lambda(\xi_i) + \lambda(\xi_t))) e_{it} \mathbf{x}_t(p) + \beta \mathbf{a}_p^T \mathbf{z}_{p,i}}{\frac{\gamma}{s^2} \sum_{t \neq i}^N (v_i v_t - w_{it} (\lambda(\xi_i) + \lambda(\xi_t))) e_{it} + \beta}, \quad (25)$$

where s is the scale parameter of the Gaussian kernel, $\mathbf{v} = \mathbf{C}^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f)$, $w_{ij} = (\mathbf{I} + 2\sigma\boldsymbol{\Lambda} + 2\gamma \mathbf{K}_{\mathbf{X}} \boldsymbol{\Lambda})^{-1}$, and $e_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2s^2)$.

To configure the proximity operator penalty, β , we multiply Eq. (24) by $\frac{\gamma + \beta}{\gamma + \beta}$ and define $\tau = \frac{\beta}{\gamma + \beta}$. Hence, Equation (24) can then be written as

$$\mathbf{X}^T = \left[-(1 - \tau) \mathbf{C}^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f) (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f)^T \mathbf{C}^{-1} + (1 - \tau) (\mathbf{C}^{-1} \boldsymbol{\Sigma}_f 2\boldsymbol{\Lambda}) + \tau \mathbf{I}_N \right]^{-1} \tau \mathbf{Z}^T \langle \mathbf{A} \rangle^T. \quad (26)$$

Note that $\tau \in [0, 1]$ and when $\tau = 1$, we obtain $\mathbf{X} = \langle \mathbf{A} \rangle \mathbf{Z}$ in Eq. (26). For Gaussian kernels we proceed in the same manner but multiplying Eq. (25) by $\frac{\gamma/s^2 + \beta}{\gamma/s^2 + \beta}$.

Let us now summarize the estimation procedure. Starting with $\mathbf{X}^0 = \mathbf{A}^0 \mathbf{Z}$, $(\mathbf{K}_{\mathbf{X}}^0)_{ij} = \mathbf{k}(\mathbf{x}_i^0, \mathbf{x}_j^0)$, $\mathbf{a}_i^0 = \text{identity filter}$, $\alpha_i^0 = 1$, $\boldsymbol{\mu}^0 = 0$, $\boldsymbol{\Sigma}^0 = \mathbf{I}$, $\boldsymbol{\Sigma}_f^0 = \gamma^0 \mathbf{K}_{\mathbf{X}}^0 + \sigma^0 \mathbf{I}$, and $\boldsymbol{\xi}_i^0 = 1$, the method iterates until convergence between Eqs. (14), (16), (17), (18), (20), (21), (22) and (23). We use the old value of the parameter in the right hand side of the estimations to obtain the new values in the left hand side. For the value for τ , we have experimentally found that using $\tau^{n+1} = \min(\tau^n + 0.001, 1)$ made the iterative process to first concentrate on the estimation of the model parameters and then proceed to estimate the filter coefficients. See the experimental section to determine the initial value of τ . At convergence of the estimation procedure we obtain the classifier and the filterbank coefficients.

4. CLASSIFICATION OF NEW PIXELS

In order to classify a new sample \mathbf{z} we transform it using the equation $\mathbf{x} = \langle \mathbf{A} \rangle \mathbf{z}$ where $\langle \mathbf{A} \rangle$ has been obtained at convergence of the training phase and denote by $f_{\mathbf{x}}$ its associated latent variable. Then $p(f_{\mathbf{x}} | \mathbf{f}, \mathbf{X}, \mathbf{x}, \boldsymbol{\mu}, \gamma, \sigma)$ is a Gaussian distribution with mean and variance

$$\langle f_{\mathbf{x}} | \mathbf{f}, \mathbf{X}, \mathbf{x}, \boldsymbol{\mu}, \gamma, \sigma \rangle = \boldsymbol{\mu} + \mathbf{h}^T \mathbf{C}^{-1} (\mathbf{f} - \boldsymbol{\mu}_f),$$

$$\text{var}(f_{\mathbf{x}} | \mathbf{f}, \mathbf{X}, \mathbf{x}, \boldsymbol{\mu}, \gamma, \sigma) = c - \mathbf{h}^T \mathbf{C}^{-1} \mathbf{h},$$

where $c = \gamma \mathbf{k}(\mathbf{x}, \mathbf{x})$ and $\mathbf{h} = \gamma (\mathbf{k}(\mathbf{x}, \mathbf{x}_1), \dots, \mathbf{k}(\mathbf{x}, \mathbf{x}_N))^T$ and γ , σ and $\boldsymbol{\mu}$ have been provided by the proposed method at convergence.

We then have

$$p(f_{\mathbf{x}} | \mathbf{y}, \mathbf{X}, \mathbf{x}, \boldsymbol{\mu}, \gamma, \sigma) = \int_{\mathbf{f}} p(f_{\mathbf{x}} | \mathbf{f}, \mathbf{X}, \mathbf{x}, \boldsymbol{\mu}, \gamma, \sigma) \mathcal{N}(\mathbf{f} | \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f) d\mathbf{f},$$

which is a Gaussian distribution with parameters

$$\langle f_{\mathbf{x}} | \mathbf{y}, \mathbf{X}, \mathbf{x}, \boldsymbol{\mu}, \gamma, \sigma \rangle = \boldsymbol{\mu} + \mathbf{h}^T \mathbf{C}^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f),$$

$$\text{var}(f_{\mathbf{x}} | \mathbf{y}, \mathbf{X}, \mathbf{x}, \boldsymbol{\mu}, \gamma, \sigma) = \mathbf{h}^T \mathbf{C}^{-1} \boldsymbol{\Sigma}_f \mathbf{C}^{-1} \mathbf{h} + c - \mathbf{h}^T \mathbf{C}^{-1} \mathbf{h},$$

This leads to the following classification procedure

$$y_{\mathbf{x}} = \begin{cases} 1 & \text{if } \boldsymbol{\mu} + \mathbf{h}^T \mathbf{C}^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f) \geq 0 \\ 0 & \text{if } \boldsymbol{\mu} + \mathbf{h}^T \mathbf{C}^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_f) < 0 \end{cases}. \quad (27)$$

5. EXPERIMENTAL RESULTS

In this section synthetic and real experiments are conducted to evaluate the performance of the proposed method, named GPF. In both experiments, we used filters of size 3×3 , 5×5 , 7×7 and 9×9 . We ran the proposed method for different values of τ^0 in the interval $[0.1, 0.9]$ with step 0.01 and selected the one giving the best classification results. GPF was compared with the SVMF method [13] which jointly learn a SVM classifier and estimates a filterbank as well as a GP classifier which does not filter the data. To do this, SVM objective function is augmented with a regularization term on the filters. Hence, in addition to the cost parameter of the SVM (C), the regularization coefficient of the filter (λ) has to be selected. We ran SVMF on $(C, \lambda) \in \{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 25, 50, 100\}^2$ and selected the values producing the best performance.

To obtain unbiased conclusions from the results, ten independent repetitions of the experiments were carried out. For each of them, a training set of 40 randomly selected samples (20 from each class) and a test set of 2000 samples were used. The Overall Accuracy (OA), the estimated Cohen's kappa statistic (κ -index) and Z-score are used as measures of accuracy and class agreement. We also report the computational cost in seconds of each algorithm, implemented using MATLAB[®] on a i7 at 2.80 GHz.

5.1. Synthetic data experiment

In the synthetic data experiment, we generated a 500×500 binary image where black and white pixels alternate in a checkerboard fashion. Observations in the class \mathcal{C}_0 (black pixels) are generated by a Gaussian distribution of mean 0.25 and standard deviation 0.4, observations of pixels in the class \mathcal{C}_1 (white pixels) are generated by a Gaussian distribution of mean 0.75 and standard deviation 0.4. Figure 1a shows a zoom of the observation dataset. Notice that it is hard to decide the class of some pixels by considering only their values.

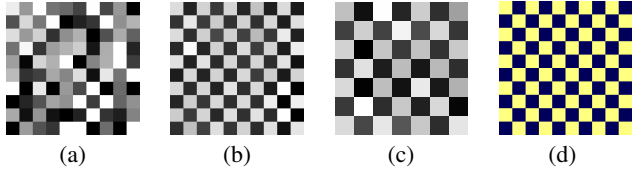


Fig. 1. (a) A set of observations of the synthetic dataset. (b) Filtered observations with the estimated kernel. (c) Estimated 7×7 kernel. (d) Classification map: dark blue C_0 , light yellow C_1 .

Table 1. Figures of merit for the synthetic experiment.

Sizes	GPF					SVMF					
	τ^0	OA	κ	Z	Time	C	λ	OA	κ	Z	Time
3×3	0.86	96.06	0.9212	106.75	0.26	0.1	5	95.84	0.9168	103.80	0.30
5×5	0.86	99.72	0.9943	452.77	0.31	0.5	1	99.69	0.9938	444.85	0.56
7×7	0.87	100	1	∞	1.02	0.5	1	100	1	∞	0.50
9×9	0.87	100	1	∞	1.69	0.5	1	100	1	∞	0.60
No Filter	-	71.06	0.4314	21.39	0.05	0.1	-	72.72	0.4537	22.74	0.003

However, in the filtered image, shown in Fig. 1b, it is easier to distinguish the class of each pixel. It is worth noting that the estimated filter, depicted in Fig. 1c, alternates positive coefficients, in the position of pixels belonging to the class of kernel central pixel, with negative coefficient, in the remainder positions. Figure 1d displays the classification map for the image in Fig. 1a, with a 100% OA.

Mean values for OA, κ -index and Z-score and the value of τ providing the best classification results are reported in Table 1. The proposed GPF method obtained an OA above 96% for all considered filter sizes and, for sizes of 7×7 and 9×9 , the estimated filter is capable to linearly separate both class and a 100% OA is obtained. In all the cases, an improvement of almost 30% is obtained over the base case where the data are not filtered (see the last row of Table 1). The computational cost of the algorithm is very limited needing only between 0.26 and 1.69 seconds to perform both training and classification tasks. The figures of merit for the SVMF method are very similar to those of the GPF although the proposed method scored slightly better for the kernel sizes of 3×3 and 5×5 .

5.2. Real data experiment

The dataset was extracted from a 7-bands satellite image of city of Naples (Italy) captured by the Landsat TM sensor in 1995 in the Urban Expansion Monitoring project (UrbEx) [16]. A small RGB region of this image is displayed in Fig. 2a. A reference land cover map was also provided by the Italian Institute of Statistics (ISTAT). The goal is the discrimination of urban (C_1) versus non-urban (C_0) land-cover classes. The reference land cover map for the image in Fig. 2a is shown in Fig. 2b. Light yellow color represents urban class, dark blue color represents non-urban and red corresponds to pixels whose class is unknown.

In this experiment we used a Gaussian kernel with parameter $s = 100$. This value was selected as the one giving the best results for SVMF. Table 2 shows the mean values for OA, κ -index and Z-score for GPF and SVMF method. Baseline case results, when no filtering is used, are also reported. GPF obtained over a 95% OA, values above 0.90 of κ -index and Z-score values close to 100 for all filter sizes, while the running time moved from 0.48 to 1.62 seconds as the kernel size increased. Those figures of merit indicates a small but significant improvement over the baseline case. In this real situation, GPF consistently obtained better results than SVMF for all kernel sizes. Also, the proposed GPF method ran much faster than

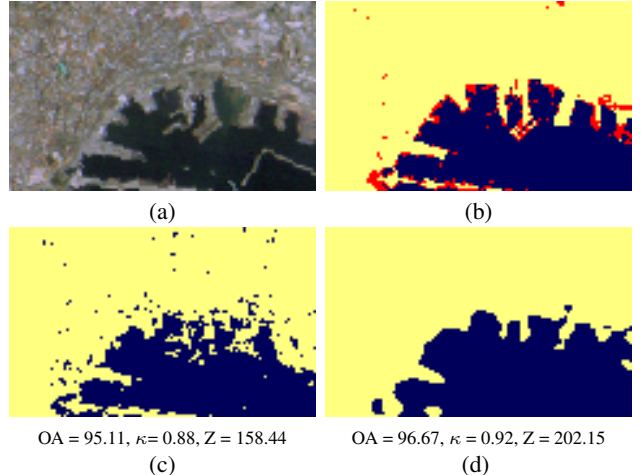


Fig. 2. (a) RGB representation of a small region of the real image. (b) Its reference land cover map. (c) Classification map without filtering. (d) Classification map with filtering.

Table 2. Figures of merit for the real experiment.

Sizes	GPF					SVMF					
	τ^0	OA	κ	Z	Time	C	λ	OA	κ	Z	Time
3×3	0.89	95.18	0.9036	95.02	0.48	50	100	93.92	0.8785	86.02	6.25
5×5	0.87	95.78	0.9156	103.03	0.64	100	50	93.16	0.8633	77.55	20.22
7×7	0.90	95.64	0.9127	100.42	0.81	100	50	93.09	0.8618	76.93	51.15
9×9	0.85	95.25	0.9049	95.50	1.62	100	50	92.88	0.8558	74.61	79.81
No Filter	-	93.21	0.8542	77.15	0.34	0.01	-	92.88	0.8577	74.93	0.11

SVMF (more than 50 times faster in some cases).

To better understand the role of filtering in the proposed method, Figures 2c and 2d depict the classification map for the image in Fig. 2a when no filtering was applied and when kernels of size 5×5 were estimated, respectively. The classification map when the image is not filtered is quite noisy, specially at the boundary of urban and non-urban areas, while the one for the filtered image exhibit more homogeneous regions and, although some pixels are misclassified, class boundaries are much better delimited. The figures of merit for this particular area are shown under their corresponding map. Although the OA for the filtered case is only a 1.5% better than the one for the unfiltered case, filtering allows for a significantly better class agreement reflected in a higher κ -index and Z-score.

6. CONCLUSIONS

In this work we have presented a new method to jointly filter and classify a signal or an image. Using Bayesian modeling and variational inference we have developed an iterative procedure to jointly estimate the classifier parameters, the filterbank and the model parameters. In the experimental section we have shown that the estimated filters helps to improve the classifier performance. The proposed method has been compared with other classification/filtering approaches, and experimental results have shown that the proposed method is more accurate and efficient.

7. REFERENCES

- [1] I. Amro, J. Mateos, M. Vega, R. Molina, and A.K. Katsaggelos, "A survey of classical methods and new trends in pansharpen-

- ing of multispectral images,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, pp. 2011:79, September 2011.
- [2] P. Ruiz, J. Mateos, G. Camps-Valls, R. Molina, and A.K. Katsaggelos, “Bayesian active remote sensing image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2186–2196, 2014.
 - [3] B. Amizic, L. Spinoulas, R. Molina, and A. K. Katsaggelos, “Compressive sampling with unknown blurring function: application to passive millimeter-wave imaging,” in *IEEE International Conference on Image Processing*. Orlando, Florida, October 2012, pp. 925–928.
 - [4] O. Martinez, L. Ferraz, X. Binefa, I. Gomez, and C. Dorronsoro, “Concealed object detection and segmentation over millimetric waves images,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2010, pp. 31–37.
 - [5] D. Gutiérrez and D. I. Escalona-Vargas, “EEG data classification through signal spatial redistribution and optimized linear discriminants,” *Computer Methods and Programs in Biomedicine*, vol. 97, no. 1, pp. 39–47, 2010.
 - [6] S. Ryali, K. Supekar, T. Chen, and V. Menon, “Multivariate dynamical systems models for estimating causal interactions in fMRI,” *NeuroImage*, vol. 54, no. 2, pp. 807–823, 2011.
 - [7] N. D. Lawrence, “Gaussian process latent variable models for visualisation of high dimensional data,” in *Advances in Neural Information Processing Systems*, 2004.
 - [8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Discriminative learned dictionaries for local image analysis,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, 2008, pp. 1–8.
 - [9] I. Ramirez, P. Sprechmann, and G. Sapiro, “Classification and clustering via dictionary learning with structured incoherence and shared features,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010*, 2010, pp. 3501–3508.
 - [10] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in *The Handbook of Brain Theory and Neural Networks*, Michael A. Arbib, Ed., pp. 255–258. MIT Press, 1998.
 - [11] B. De Vries and J. C. Principe, “The gamma model—A new neural model for temporal processing,” *Neural Networks*, vol. 5, pp. 565–576, 1992.
 - [12] S. Lawrence, A. C. Tsoi, and A. D. Back, “The gamma MLP for speech phoneme recognition,” in *Advances in Neural Information Processing Systems*. 1996, pp. 785–791, MIT Press.
 - [13] R. Flamary, D. Tuia, B. Labbe, G. Camps-Valls, and A. Rakotomamonjy, “Large margin filtering,” *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 648–659, 2012.
 - [14] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, 2007.
 - [15] G. Bouchard, “Efficient bounds for the softmax function, applications to inference in hybrid models,” in *2007 Neural Information Processing Systems Conference, NIPS 2007*, 2007, vol. 6239.
 - [16] P. Castracane, F. Iavarone, S. Mica, E. Sottile, C. Vignola, and O. Arino, “Monitoring urban sprawl and its trends with EO data. UrbEx, a prototype national service from a WWF-ESA joint effort,” in *2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, 2003, pp. 245–248.