PASSIVE MILLIMETER WAVE IMAGE CLASSIFICATION WITH LARGE SCALE GAUSSIAN PROCESSES

Pablo Morales[1], *Adrián Pérez-Suay*[2], *Rafael Molina*[1], *Gustau Camps-Valls*[2], *Aggelos K. Katsaggelos*[3]

[1]Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain
[2]Image Processing Laboratory (IPL), Universitat de València, Spain
[3]Dept. of Electrical Engineering and Computer Science, Northwestern University, USA

ABSTRACT

Passive Millimeter Wave Images (PMMWIs) are being increasingly used to identify and localize objects concealed under clothing. Taking into account the quality of these images and the unknown position, shape, and size of the hidden objects, large data sets are required to build successful classification/detection systems. Kernel methods, in particular Gaussian Processes (GPs), are sound, flexible, and popular techniques to address supervised learning problems. Unfortunately, their computational cost is known to be prohibitive for large scale applications. In this work, we present a novel approach to PMMWI classification based on the use of Gaussian Processes for large data sets. The proposed methodology relies on linear approximations to kernel functions through random Fourier features. Model hyperparameters are learned within a variational Bayes inference scheme. Our proposal is well suited for real-time applications, since its computational cost at training and test times is much lower than the original GP formulation. The proposed approach is tested on a unique, large, and real PMMWI database containing a broad variety of sizes, types, and locations of hidden objects.

***Index Terms*—** Gaussian processes, large scale classification, random Fourier features, variational inference, PMMWI.

1. INTRODUCTION

Millimeter wave images can be used to recognize and localize hidden objects under clothing [1]. This type of images is becoming increasingly popular in threat detection systems located at warehouses and airports (such as international ones in Los Angeles or San Francisco). In contrast to *active* scanners, which direct millimeter wave radiation to the subject and then interpret the reflected energy, *passive* systems use only ambient radiation and that emitted from the human body or objects [2]. This means safer and less invasive systems, which are better suited for practical applications.

Sensor modelling and image processing techniques have been used on PMMWIs. The main concepts related to millimeter images are introduced in [2, 3]. Compressive sensing and superresolution are explored in [4, 5]. Denoising, deconvolution, and enhancement techniques have also been applied to this kind of images [6–9].

Unfortunately, the literature on classification using these images is still scarce. K-means is used to segment PMMWIs in [10]. Gaussian mixture models are applied in [11] to characterize people with and without threats. In [12] the authors apply noise elimination and image segmentation using Local Binary Fitting (LBF). A highly efficient two-step algorithm, based on denoising and mathematical morphology, was proposed in [13]. It achieves an acceptable detection rate on noisy and low contrast images. A comprehensive comparison between classical methods (Logistic Regression, SVM, Random Forest, Boosting) is provided in [14]. In that work, the large size of the data set imposed constraints on the kernel used for SVM, and prevented the application of GPs.

Kernel machines [15, 16] are among the most popular approaches for supervised learning. Due to its solid Bayesian treatment, GP [17] is a current state-of-the-art kernel method, which has also been used for image classification [18, 19]. For a problem with n training instances, kernel machines store and manipulate kernel matrices of size $n \times n$, which makes them scale as $\mathcal{O}(n^3)$ in training and $\mathcal{O}(n^2)$ for each test instance. These two orders hamper their applicability to large scale problems in terms of computational limitations and impossibility of real-time prediction respectively.

In this paper we develop a new method that allows for large scale PMMWI Gaussian Processes classification. Our approach relies on linear approximations to kernel matrices based on random Fourier features (RFF) [20]. The use of GPs with RFF for regression problems has been presented in [21–23]. To deal with the non-conjugate observation model typical of classification problems, we resort here to the variational inference approach [24, Section 10.6]. Computational complexity at test turns out to be independent on the training set size. Cost at training is also much lower than original GP. These capabilities make our approach suitable for real-time applications.

The rest of the paper is organized as follows. The proposed method is derived from standard GP theory and random Fourier features in Section 2. Section 3 presents our PMMWI data set, its preprocessing, and the experimental setting. Section 4 shows competitive empirical results for our approach, which outperforms direct application of GPs and allows for real-time prediction. Section 5 concludes the paper with some remarks and future outlook.

2. GAUSSIAN PROCESSES AND RANDOM FOURIER FEATURES

Gaussian Processes (GPs) [17] are popular Bayesian models for supervised machine learning tasks, such as regression and classification. In the latter case, we are provided with an input-output data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. Following a function-space view, GPs codify this relationship by means of latent variables $\{f_i = f(\mathbf{x}_i) \in \mathbb{R}\}_{i=1}^n$. These variables are jointly normally distributed as $\mathcal{N}(\mathbf{0}, (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i,j \leq n})$, with $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ the kernel function. The outputs y_i depend on f_i by means of the sig-

Research funded by the European Research Council (ERC) under the ERC-CoG-2014 SEDAL project (grant agreement 647423), the Spanish Ministry of Economy and Competitiveness (MINECO) projects TIN2013-43880-R, TIN2015-64210-R, and DPI2016-77869-C2-2-R, and a Catalyst award, McCormick School of Engineering, Northwestern University.

moid observation model $\mathrm{p}(y_i|f_i) = \psi(f_i) = (1+\exp(-f_i))^{-1}$.

As any other kernel method, GP for classification (GPC) is computationally prohibitive when dealing with large scale problems. The training step scales as $\mathcal{O}(n^3)$, whereas the computational complexity of the prediction is $\mathcal{O}(n^2)$ for each test instance [17]. In the case of a standard desktop computer, this places its computational feasibility limit around $n = 10^4$ training examples.

2.1. Random Fourier features

The work [20] presents a general methodology (based on Bochner's theorem [25]) for theoretically approximating a positive-definite shift-invariant kernel k. This is achieved by explicitly projecting the original d-dimensional data onto $\mathcal{O}(D)$ random Fourier features, whose standard linear kernel approximates k. This linearity allows us to reduce $n \times n$ matrix inversions to $\mathcal{O}(D) \times \mathcal{O}(D)$ ones, which decreases the computational cost to $\mathcal{O}(nD^2)$ during training and to $\mathcal{O}(D^2)$ for each test instance. For large scale applications, D will be much lower than n, which implies a significant benefit.

In this work we use the well-known Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \gamma \cdot \exp(-\|\mathbf{x}-\mathbf{x}'\|^2/(2\sigma^2))$, which can be linearly approximated as $k(\mathbf{x}, \mathbf{x}') \approx \mathbf{z}(\mathbf{x})^\intercal \mathbf{z}(\mathbf{x}')$, with

$$\mathbf{z}(\mathbf{x})^\intercal = \sqrt{\gamma D^{-1}} \left(\cos\left(\sigma^{-1}\mathbf{w}_1^\intercal \mathbf{x}\right), \sin\left(\sigma^{-1}\mathbf{w}_1^\intercal \mathbf{x}\right), \dots \right.$$
$$\left. \dots, \cos\left(\sigma^{-1}\mathbf{w}_D^\intercal \mathbf{x}\right), \sin\left(\sigma^{-1}\mathbf{w}_D^\intercal \mathbf{x}\right)\right) \in \mathbb{R}^{2D}. \quad (1)$$

As indicated in [20], the error in this approximation exponentially decreases with the number D of Fourier frequencies $\mathbf{w}_i \in \mathbb{R}^d$, which must be independently sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. In matrix notation, we approximate the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ with the explicitly mapped data $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_n]^\intercal \in \mathbb{R}^{n \times D}$ as $\mathbf{K} \approx \mathbf{Z}\mathbf{Z}^\intercal$.

In order to approximate a GP classifier with Gaussian kernel, we will consider a Bayesian linear model working on these new features. Hyperparameters γ and σ in eq. (1) will be optimized within a variational approach in order to maximize the marginal likelihood of observed data. Notice that although we could include the estimation of $\mathbf{w}_1, \dots, \mathbf{w}_D$ in the Bayesian framework, we concentrate here on the computing capabilities of the proposed approach and assume the frequencies are sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and fixed.

2.2. Modelling and Inference

We consider the standard binary-classification logistic regression model defined over the explicitly mapped Fourier features $\mathbf{z}_i$:

$$\mathrm{p}_{\boldsymbol{\theta}}(y_i = 1|\boldsymbol{\beta}) = (1+\exp(-\boldsymbol{\beta}^\intercal \mathbf{z}_i))^{-1}, \quad (2)$$

where $\boldsymbol{\theta} = (\theta_1 = \sqrt{\gamma}, \theta_2 = \sigma)$ and $\mathbf{z}_i$ is given by eq. (1).

Weights $\boldsymbol{\beta} \in \mathbb{R}^{2D}$ are assigned a normal prior distribution $\mathrm{p}(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}|\mathbf{0}, \mathbf{I})$. Thus, the joint p.d.f. reads

$$\mathrm{p}_{\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\beta}) = \mathrm{p}_{\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\beta})\mathrm{p}(\boldsymbol{\beta}). \quad (3)$$

To obtain the maximum likelihood (ML) estimate of $\boldsymbol{\theta}$, $\overline{\boldsymbol{\theta}}$, we integrate the above joint distribution on $\boldsymbol{\beta}$ and maximize on $\boldsymbol{\theta}$ the marginal distribution $\mathrm{p}_{\boldsymbol{\theta}}(\mathbf{y})$. The posterior distribution $\mathrm{p}_{\overline{\boldsymbol{\theta}}}(\boldsymbol{\beta}|\mathbf{y})$ is then calculated. However, the sigmoid likelihood in eq. (2) makes these computations analytically intractable, and we resort to the *variational* inference approximation [24, Section 10.6].

First, we use the variational bound [24]

$$\log\left(1+e^x\right) \leq \lambda(\xi)(x^2-\xi^2) + \frac{x-\xi}{2} + \log\left(1+e^{\xi}\right), \quad (4)$$

where $x, \xi \in \mathbb{R}$ and $\lambda(\xi) = (1/2\xi)\left((1+e^{-\xi})^{-1} - 1/2\right)$. For a fixed x, it is easy to check that this bound is minimum when $\xi^2 = x^2$. This produces the following lower bound for the joint probability distribution in eq. (3):

$$\log \mathrm{p}_{\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\beta}) \geq -\frac{1}{2}\boldsymbol{\beta}^\intercal \left(\mathbf{Z}^\intercal(2\boldsymbol{\Lambda})\mathbf{Z} + \mathbf{I}\right)\boldsymbol{\beta} + \mathbf{v}^\intercal \mathbf{Z}\boldsymbol{\beta} + C(\xi_i), \quad (5)$$

where $\mathbf{v} = \mathbf{y} - (1/2)\cdot\mathbf{1}_{n\times 1}$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda(\xi_1), \dots, \lambda(\xi_n))$.

Notice that this lower bound is quadratic in $\boldsymbol{\beta}$, which enables us to analytically work with it. Namely, using the exponential function on eq. (5), integrating out $\boldsymbol{\beta}$, and maximizing on $\boldsymbol{\theta}$ (recall $\mathbf{Z} = \mathbf{Z}(\boldsymbol{\theta})$), we obtain the following approximation of $\overline{\boldsymbol{\theta}}$:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left(\mathbf{v}^\intercal \mathbf{Z}\left(\mathbf{Z}^\intercal(2\boldsymbol{\Lambda})\mathbf{Z} + \mathbf{I}\right)^{-1}\mathbf{Z}^\intercal \mathbf{v} - \log\left|\mathbf{Z}^\intercal(2\boldsymbol{\Lambda})\mathbf{Z} + \mathbf{I}\right|\right). \quad (6)$$

To obtain the posterior $\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{\beta}|\mathbf{y})$, we fix $\boldsymbol{\theta}$ in eq. (3) and find the same problem with the sigmoid function. We again resort to the quadratic variational bound given by eq. (5), which provides the following approximate posterior normal distribution for $\boldsymbol{\beta}$:

$$\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{\beta}|\mathbf{y}) \approx \mathrm{q}_{\boldsymbol{\theta}}(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}),$$
$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \left(\mathbf{Z}^\intercal(2\boldsymbol{\Lambda})\mathbf{Z} + \mathbf{I}\right)^{-1}, \quad \boldsymbol{\mu}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}}\mathbf{Z}^\intercal \mathbf{v}. \quad (7)$$

Finally, for a given $\boldsymbol{\theta}$, the bound in eq. (5) is optimal when

$$\xi_i^2 = \langle(\boldsymbol{\beta}^\intercal \mathbf{z}_i)^2\rangle_{\mathrm{q}_{\boldsymbol{\theta}}(\boldsymbol{\beta}|\mathbf{y})} \quad \forall i = 1, \dots, n, \quad (8)$$

which yields

$$\xi_i = \sqrt{(\mathbf{z}_i^\intercal \boldsymbol{\Sigma}_{\boldsymbol{\beta}}\mathbf{Z}^\intercal \mathbf{v})^2 + \mathbf{z}_i^\intercal \boldsymbol{\Sigma}_{\boldsymbol{\beta}}\mathbf{z}_i} \quad \forall i = 1, \dots, n. \quad (9)$$

Our method is named RFF-GPC. At training time, it is fed with labelled data and runs iteratively to calculate $\hat{\boldsymbol{\theta}}$ and the approximate normal posterior $\mathrm{q}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\beta})$ (see Algorithm 1).

Algorithm 1 Training RFF-GPC

Input: Data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{0, 1\}$.
Randomly sample Fourier frequencies $\mathbf{w}_i$ in eq. (1) from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.
Initialize $\mathrm{q}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\beta})$ to $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (that is, its prior distribution).
Initialize $\hat{\boldsymbol{\theta}}$
repeat
  Update each ξ_i using eq. (9).
  Update $\hat{\boldsymbol{\theta}}$ using eq. (6).
  Update the posterior $\mathrm{q}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\beta})$ using eq. (7).
until convergence
return Optimal estimator $\hat{\boldsymbol{\theta}}$ and the posterior $\mathrm{q}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\beta})$.

At test time, the probability of class 1 for a previously unseen instance $\mathbf{x}_* \in \mathbb{R}^d$ is:

$$\mathrm{p}(y_* = 1) \approx \int \mathrm{p}_{\hat{\boldsymbol{\theta}}}(y_* = 1|\boldsymbol{\beta})\mathrm{q}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\beta})\mathrm{d}\boldsymbol{\beta} \approx$$
$$\approx \psi\left(\mathbf{z}_*^\intercal \boldsymbol{\mu}_{\boldsymbol{\beta}} \cdot (1 + (\pi/8)\mathbf{z}_*^\intercal \boldsymbol{\Sigma}_{\boldsymbol{\beta}}\mathbf{z}_*)^{-1/2}\right),$$

with ψ the sigmoid function. The computations involved in these algorithms scale as $\mathcal{O}(nD^2)$ for training stage, and $\mathcal{O}(D^2)$ for prediction on each test point. Unlike standard GPC, the latter is independent on the number of training instances. This significant reduction makes our proposal suitable for large scale real-time applications.

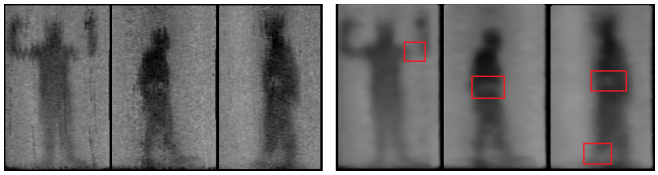

Fig. 1: Examples of PMMWIs with hidden threats before preprocessing (left) and after it (right). Red boxes on the right highlight object locations, which correspond to lighter areas of the body.

3. IMAGE PREPROCESSING AND EXPERIMENTAL SETTING

3.1. General preprocessing

We have available a PMMWIs database of 3309 images of size 125×195. They show people in different positions who may hide (up to two) objects of varied sizes and shapes at different locations. Unfortunately, acquired images suffer from spatially variant noise, and their general quality is poor. These problems specially manifest at the contour of the individuals, which can be confused (even by a human thorough gaze) with hidden threats. This prevents classical models from achieving competitive detection results, see also [14]. To better identify threats, the image signal to noise ratio must be increased, and the contrast enhanced. This is addressed by a combination of linear (local mean) and non-linear (local median) smoothing filters. Figure 1 shows examples of raw and processed images in the database (see [14] for more images).

Given a new image, our goal is to automatically discern whether it contains some threat or not. We will not fit the machine learning algorithms directly over the global images, but extract relevant features from local patches. Namely, for every 2×2 non-overlapping block we consider three regions of areas 39×39, 19×19, and 9×9 (these areas have been selected taking into account the targeted threat sizes, neither too big nor too small) centered at one of its four points. We only consider *active* pixels, i.e. those whose 39×39 region is completely contained in the image. To each one of these blocks we assign a feature vector constructed by concatenating Haar traits [26] extracted from its three associated regions. On each region we use 115 Haar filters, which are appropriately chosen for the shapes of hidden objects. This yields a 345-dimensional feature vector for each block, which are considered positive (having threat) if their 39×39 region overlaps a hidden object by at least 50%, see [14] for additional details. In the sequel, we will refer to these blocks (together with their feature vector) as learning *instances*. In total, we have 3476 such instances for each image.

3.2. Experimental setting

To test the proposed model we perform a five folds cross validation. We ensure that each partition contains the same number of images with none, one, and two threats. In the training data sets we only include a positive instance if it (more precisely, its 39×39 associated region) completely covers the threat. Moreover, since adjacent negative instances have equivalent feature vectors, we only keep one from every 2×2 group of them. Table 1 summarises the number of available training instances for the experiments.

Table 1 also reveals that the number of negative instances is approximately six times the number of positive ones in each training data set. This imbalance could bias the algorithms against the minority class. To avoid it we train six classifiers per fold, each one using all positive instances and a random sample (without replacement and of equivalent size) of the negative instances in that fold. This means a training data set with $n = 6 \cdot 10^5$ instances approximately.

Table 1: Number of negative and positive instances in the training data set for each fold.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Negative	1896222	1895130	1898030	1897209	1898825
Positive	299838	302417	299829	299644	298212
Total	2196060	2197547	2197859	2196853	2197037

Standard GPC cannot cope with this large scale setting. In order to comparatively evaluate their performance against RFF-GPC, we consider a (balanced) subsample of size $n = 10^4$ for each experiment, thus taking GPC until its computational limit. Once the six models are trained, predictions (i.e. probabilities of class 1) for test instances are collected and the mean is computed.

For GPC we make use of the full 345-dimensional feature vectors. However, in the case of RFF-GPC we will need to previously reduce this dimensionality.

3.3. Preprocessing for RFF-GPC

A key parameter in our method is the number D of random Fourier features used. It can be checked, both empirically and theoretically [20, Claim 1], that the approximation $\mathbf{K} \approx \mathbf{Z}\mathbf{Z}^\intercal$ exponentially improves with D. Likewise, it exhibits an exponential deterioration with the number d of original features in the data set [20].

In practice, for $d = 345$ like in our case, we would need a value of D greater than 10^5. However, D should not exceed $5 \cdot 10^3$ if we want the training $\mathcal{O}(nD^2)$ to be computationally feasible in a standard desktop computer. This would make RFF-GPC even more computationally prohibitive than standard Gaussian Processes.

In order to overcome this problem, we need to significantly reduce the dimensionality of the data set before applying RFF-GPC, but preserving as much information as possible. For this, we resort to Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) (see [27] for an interesting comparison). The first one extracts projections that keep most of the variability of the original data [24, Chapter 12], whereas the latter provides features that best linearly separate both classes [24, Section 4.1.4]. We decided to include 15 principal components (a PCA analysis reveals that they already explain the 97.69% of original variance) and 5 linear discriminant directions. Therefore, we reduce the dimensionality to $d' = 20$. With this, we can consider more reasonable values for D, which will be fixed at $D = 500$ in our experiments. In Sections 4 and 5 we discuss future work related to this parameter.

4. EXPERIMENTAL RESULTS

First, we compare prediction time between RFF-GPC and GPC. Figure 2 shows elapsed time to predict one image, i.e. 3476 patches. We show the evolution in terms of training data set size. To do this, we extracted balanced subsamples from the largest training data set in each experiment. We observe that, whereas GPC clearly exhibits an increasing dependence on the training set size, RFF-GPC is not affected by this quantity. This is the expected behaviour from theoretical test orders $\mathcal{O}(n^2)$ and $\mathcal{O}(D^2)$ respectively. RFF-GPC outperforms GPC at any training size, being more than 100 times faster at the full models ($n = 6 \cdot 10^5$ and $n = 10^4$ respectively).

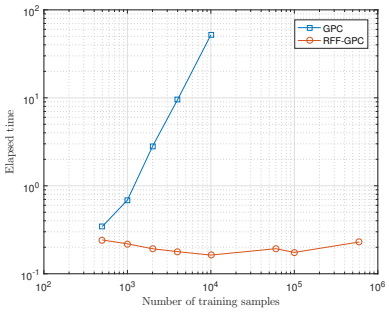

Fig. 2: Elapsed prediction time per image for GPC and RFF-GPC. Notice the logarithmic scale in both axes.

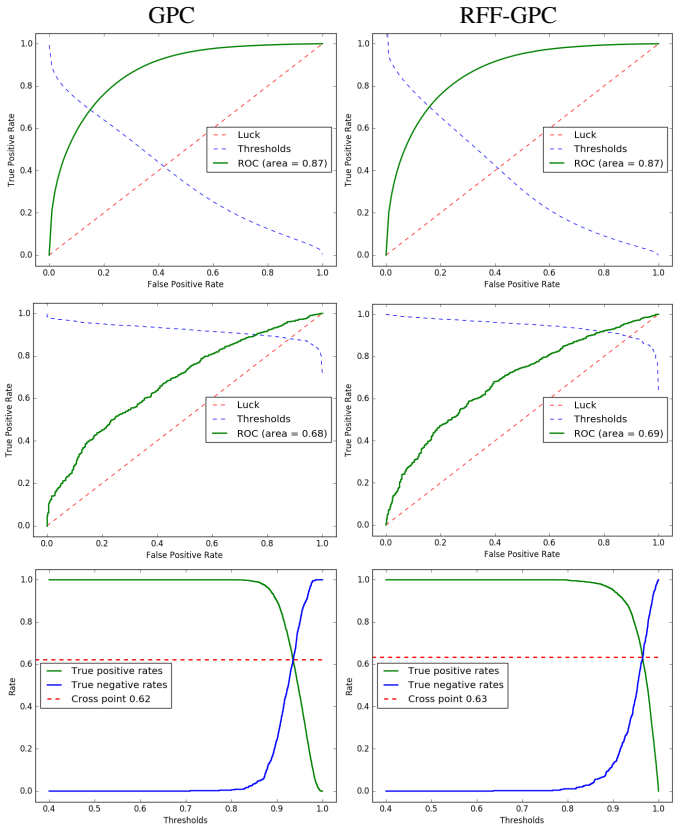

Fig. 3: Comparative classification results for GPC (left column) and RFF-GPC (right column). First and second rows show the ROC curves (and AUC) at patch and image levels respectively. Third row explictly depicts the evolution of TPR, TNR, and their cross point.

The second experiment compares the classification performance of GPC and RFF-GPC at both patch and image levels. For the latter, each test image is associated to the highest probability of threat among all its patches. We use the full models, i.e. $n = 10^4$ for GPC and $n \approx 6 \cdot 10^5$ for RFF-GPC. In Figure 3 we show the ROC curves at patch and image levels. They represent the trade-off between true positive (TP) and false positive (FP) rates when considering different thresholds over the test probabilities of class 1 (having threat). The TPR-TNR evolution and their cross point at image level are also provided. We observe that both methods behave similarly, with RFF-GPC slightly outperforming GPC at image level.

We also carried out a preliminary experiment to assess the quality of our approach when using a higher number of random Fourier features D (see future work in Section 5). We used RFF-GPC with $D = 1750$ and $n = 1.5 \cdot 10^5$ training instances (both values can be still further increased). This significantly improves previous results, see Figure 4. Namely, AUC metric and TPR-TNR cross point improve from 0.69 and 0.63 to 0.72 and 0.65 respectively. These results suggest that increasing and wisely combining the values of D and n is a promising future work.

It is worth noticing that our novel approach is competitive with the state-of-the-art results over this data set presented in [14]. The new method beats other kernel machines like SVM, and only Random Forest (RF) performs slightly better. Namely, at image level, TP and TN rates cross at 0.68 for RF (see [14, Figure 6]) while, as we have already indicated, the crossing point for our method is at 0.65. The future use of prior models on the Fourier frequencies $\mathbf{w}_i$ and higher values of D and n are expected to surpass RF performance.

To sum up, our results are competitive with state-of-the-art ones. We already outperform classical GPC allowing for a much faster prediction (which is a remarkable benefit for real-world applications). More complex modelling of the classification problem will certainly lead to better performance.

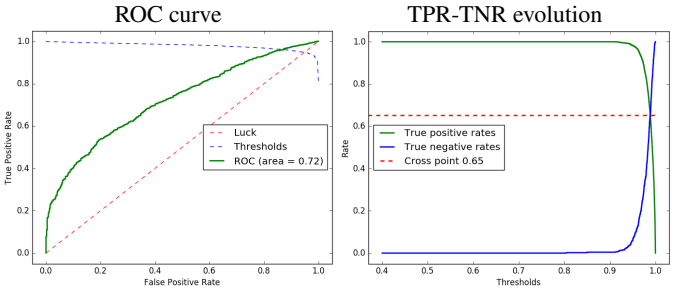

Fig. 4: ROC curve (and AUC) and TPR-TNR evolution at image level for RFF-GPC with higher value of D.

5. CONCLUSIONS AND FUTURE WORK

In this work we presented a new kernel-based approach to classify PMMWIs using a large scale training data set. The method works at patch level by extracting multiscale Haar features which are then summarized using PCA and LDA projections. The huge number of samples prevents us from using classical GPC methods on the complete data set. We resort here to the use of random Fourier features to linearize a non-linear kernel. A variational Bayes scheme is used to make inference tractable. All the model parameters are estimated. The proposed approach outperforms classical GPC, it is suitable for real-time applications, and produces competitive results with current state-of-the-art methods.

Since detection results are negatively affected by the poor quality of images, we are currently working on the use of image processing techniques to improve acquired images. Furthermore, we are also investigating the optimal relationship between the number of projected features d', Fourier frequencies D, and training examples n. The optimal estimation of Fourier frequencies in a probabilistic sense is also currently under study.

6. REFERENCES

[1] D. M. Sheen, D. L. McMakin, and T. E. Hall, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Transactions on microwave theory and techniques*, vol. 49, no. 9, pp. 1581–1592, 2001.

[2] L. Yujiri, M. Shoucri, and P. Moffa, "Passive millimeter wave imaging," *IEEE Microwave Magazine*, vol. 4, no. 3, pp. 39–50, 2003.

[3] L. Yujiri, "Passive millimeter wave imaging," in *2006 IEEE MTT-S International Microwave Symposium Digest*, 2006, pp. 98–101.

[4] S. D. Babacan, M. Luessi, L. Spinoulas, A. K. Katsaggelos, N. Gopalsami, T. Elmer, R. Ahern, S. Liao, and A. Raptis, "Compressive passive millimeter-wave imaging," in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 2705–2708.

[5] W. Saafin, S. Villena, M. Vega, R. Molina, and A.K. Katsaggelos, "Compressive sensing super resolution from multiple observations with application to passive millimeter wave images," *Digital Signal Processing*, vol. 50, pp. 180–190, 2016.

[6] B. Han, J. Xiong, L. Li, J. Yang, and Z. Wang, "Research on millimeter-wave image denoising method based on contourlet and compressed sensing," in *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, 2010, vol. 2, pp. V2–471–V2–475.

[7] J. Mateos, A. López, M. Vega, R. Molina, and A. K. Katsaggelos, "Multiframe blind deconvolution of passive millimeter wave images using variational dirichlet blur kernel estimation," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2678–2682.

[8] J. Yang, J. Wang, and L. Li, "A new algorithm for passive millimeter-wave image enhancement," in *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, 2010, vol. 3, pp. V3–507–V3–511.

[9] W. Yu, X. Chen, S. Dong, and W. Shao, "Study on image enhancement algorithm applied to passive millimeter-wave imaging based on wavelet transformation," in *Electrical and Control Engineering (ICECE), 2011 International Conference on*, 2011, pp. 856–859.

[10] C.D. Haworth, B.G. Gonzalez, M. Tomsin, R. Appleby, P. Coward, A. Harvey, K. Lebart, Y. Petillot, and E. Trucco, "Image analysis for object detection in millimetre-wave images," in *Passive Millimetre-wave and terahertz imaginh and technology*, 2004, vol. 5619, pp. 117–128.

[11] C.D. Haworth, Y.R. Petillot, and E. Trucco, "Image processing techniques for metallic object detection with millimetre-wave images," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1843 – 1851, 2006.

[12] O. Martínez, L. Ferraz, X. Binefa, I. Gómez, and C. Dorronsoro, "Concealed object detection and segmentation over millimetric waves images," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, June 2010, pp. 31–37.

[13] I. Gómez Maqueda, N. Pérez de la Blanca, R. Molina, and A.K. Katsaggelos, "Fast millimetre wave threat detection algorithm," in *European Signal Processing Conference (EUSIPCO 2015)*. Nice (France), September 2015, pp. 599–603.

[14] S. López Tapia, R. Molina, and N. Pérez de la Blanca, "Detection and localization of objects in passivemillimeter wave images," in *European Signal Processing Conference (EUSIPCO 2016)*, 2016.

[15] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

[16] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

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

[18] Y. Bazi and F. Melgani, "Gaussian process approach to remote sensing image classification," *IEEE transactions on geoscience and remote sensing*, vol. 48, no. 1, pp. 186–197, 2010.

[19] P. Ruiz, R. Molina, and A.K. Katsaggelos, "Joint data filtering and labeling using gaussian processes and alternating direction method of multipliers," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3059–3072, July 2016.

[20] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2007, pp. 1177–1184.

[21] M. Lázaro-Gredilla, J. Quiñonero Candela, C.E. Rasmussen, and A. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *Journal of Machine Learning Research*, , no. Jun, pp. 1865–1881, 2010.

[22] J. Hensman, N. Durrande, and A. Solin, "Variational Fourier features for Gaussian processes," *arXiv preprint arXiv:1611.06740*, 2016.

[23] A.C. Damianou and N.D. Lawrence, "Deep Gaussian Processes," *Journal of Machine Learning Research*, vol. 31, pp. 207215, 2014.

[24] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[25] W. Rudin, *Fourier analysis on groups*, John Wiley & Sons, 2011.

[26] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Computer vision, 1998. sixth international conference on*. IEEE, 1998, pp. 555–562.

[27] A. M. Martínez and A. C. Kak, "PCA versus LDA," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001.