AN M-ARY KMP CLASSIFIER FOR MULTI-ASPECT TARGET CLASSIFICATION

Xuejun Liao, Hui Li, and Balaji Krishnapuram

Department of Electrical and Computer Engineering
Duke University

ABSTRACT

The kernel matching pursuit (KMP) algorithm is re-formulated in the framework of the theory of optimal experiments, using a weighted sum of squared errors as the loss function, and it is extended to the case of *M*-ary target classification and kernel optimization. The *M*-ary KMP classifier is applied to multi-aspect classification of moving targets based on high-range resolution (HRR) radar signatures, for which the target-sensor orientations are assumed approximately known. A multi-aspect processing method is presented based on the use of the estimates of target-sensor orientation angles. The KMP classification results for ten MSTAR targets are presented, with a comparison to corresponding results using the relevance vector machine (RVM).

1. INTRODUCTION

Matching pursuits (MP) is a well-known technique for representing a signal as a linear expansion of basis functions that are selected from a potentially redundant dictionary [1]. Based on MP, the kernel matching pursuits (KMP) algorithm was introduced in [2], wherein MP was applied to kernel basis functions and extended to use the non-squared-error loss function. KMP with a squared-error loss function is closely related to the orthogonal least square (OLS) algorithm [3-4]. In this paper we re-formulate KMP in the framework of the theory of optimal experiments [5] using a weighted sum of squared errors as the loss function, and extend it to the case of *M*-ary target classification and kernel optimization. The *M*-ary KMP classifier is applied to multi-aspect classification of moving targets using high-range resolution (HRR) radar signatures, for which the target-sensor orientations are assumed approximately known, based on, for example, the target's Doppler signatures. A multi-aspect processing method is presented based on the use of the estimates of target-sensor orientation angles. The KMP classification results for ten MSTAR targets are presented, with a comparison to the corresponding results using the relevance vector machine (RVM) [6].

2. AN M-ARY KMP CLASSIFIER

2.1 The KMP

2.1.1 Estimation of the weights

The KMP implements a set of functions of the form

$$f_n(\mathbf{x}) = \sum_{i=1}^{n} w_{n,i} K(\mathbf{c}_i, \mathbf{x}) + w_{n,0} = \mathbf{w}_n^T \boldsymbol{\phi}_n(\mathbf{x}) \tag{1}$$

where $w_{n,0}$ is the bias term, $K(\cdot,\cdot)$ is the kernel

$$\boldsymbol{\phi}_n(\cdot) = [1, K(\mathbf{c}_1,\cdot), K(\mathbf{c}_2,\cdot), ..., K(\mathbf{c}_n,\cdot)]^T \tag{2}$$

with $K(\mathbf{c}_i,\cdot)$ the kernel-induced basis function centered at $\mathbf{c}_i$,

$$\mathbf{w}_n = [w_{n,0}, w_{n,1}, w_{n,2}, \cdots, w_{n,n}]^T \tag{3}$$

are the weights that combine the basis functions in the summation, and the subscript n is used to denote the number of basis functions being used.

Assume we are given a training set $\{\mathbf{x}_i, y_i\}_{i=1}^N$ of size N, where $\mathbf{x}_i$ is the i-th input and y_i its expected output, the weighted sum of squared errors between the expected output and the KMP output given in (1) is [5]

$$\begin{aligned} e_n &= \left(1 \Big/ \sum_{i=1}^N \beta_i\right) \sum_{i=1}^N \beta_i [y_i - f_n(\mathbf{x}_i)]^2 \\ &= \left(1 \Big/ \sum_{i=1}^N \beta_i\right) \sum_{i=1}^N \beta_i [y_i - \mathbf{w}_n^T \boldsymbol{\phi}_n(\mathbf{x}_i)]^2 \end{aligned} \tag{4}$$

where β_i is a weight accounting for the importance of the i-th training sample $(\mathbf{x}_i, y_i)$. For example, $1/\beta_i$ may represent the variance of y_i given $\mathbf{x}_i$, i.e., $1/\beta_i = \mathrm{var}(y_i|\mathbf{x}_i)$. In addition, if one knows *a priori* that $\mathbf{x}_i$ is better representative of the system being modeled, this can be accounted for by assigning a larger β_i. An example of the use of β is given in Sec. 3. The unknowns in (4) are the centers $\mathbf{c}_i$ of the basis functions in $\boldsymbol{\phi}_n$, and the weights in $\mathbf{w}_n$. The determination of $\mathbf{c}_i$ will be addressed separately in Sec. 2.1.2. At the moment we suppose $\mathbf{c}_i$ and consequently $\boldsymbol{\phi}_n$ are known and aim at solving for $\mathbf{w}_n$. Then the value of $\mathbf{w}_n$ that minimizes (4) is found to be [5]

$$\mathbf{w}_n = \mathbf{M}_n^{-1} \{\beta_i \boldsymbol{\phi}_{n,i} y_i\}_i \tag{5}$$

where $\boldsymbol{\phi}_{n,i}$ is an abbreviation of $\boldsymbol{\phi}_n(\mathbf{x}_i)$, $\{\cdot\}_i \overset{Def}{=} \sum_{i=1}^N (\cdot)$, and

$$\mathbf{M}_n = \sum_{i=1}^N \beta_i \boldsymbol{\phi}_n(\mathbf{x}_i) \boldsymbol{\phi}_n^T(\mathbf{x}_i) = \{\beta_i \boldsymbol{\phi}_{n,i} \boldsymbol{\phi}_{n,i}^T\}_i \tag{6}$$

Denote by E(.) and var(.) the expectation and variance, respectively. It can be shown that when $\mathrm{E}(y_i|\mathbf{x}_i) = f_n(\mathbf{x}_i) = \mathbf{w}_n^T \boldsymbol{\phi}_n(\mathbf{x}_i)$ and $\mathrm{var}(y_i|\mathbf{x}_i) = 1/\beta_i$, (5) is the best linear unbiased estimate (BLUE) of $\mathbf{w}_n$, given $\boldsymbol{\phi}_n(\mathbf{x}_i)$ [5]. Following the convention in [5], we refer to $\mathbf{M}_n$ as the Fisher information matrix, assuming that we consider only linear estimates of $\mathbf{w}_n$, and $\boldsymbol{\phi}_n(\mathbf{x}_i)$ for $i=1,2,\ldots,N$ are given.

2.1.2 Sequential Selection of Basis Functions

An nth order KMP employs n basis functions. According to the definition in (1), the $(n+1)$-th order KMP is inductively written as

$$f_{n+1}(\mathbf{x}) = \mathbf{w}_{n+1}^T \boldsymbol{\phi}_{n+1}(\mathbf{x}) \tag{7}$$

where

$$\boldsymbol{\phi}_{n+1}(\cdot) = [1, K(\mathbf{c}_1,\cdot), K(\mathbf{c}_2,\cdot),..., K(\mathbf{c}_n,\cdot), K(\mathbf{c}_{n+1},\cdot)]^T = \begin{bmatrix} \boldsymbol{\phi}_n(\cdot) \\ \phi_{n+1}(\cdot) \end{bmatrix} \quad (8)$$

with $\phi_{n+1}(.) = K(\mathbf{c}_{n+1},\cdot)$ a new basis function centered at $\mathbf{c}_{n+1}$. The weighted sum of squared errors of the (n+1)-th order KMP is

$$e_{n+1} = (1/\sum_{i=1}^{N}\beta_i)\sum_{i=1}^{N}\beta_i[y_i - f_{n+1}(\mathbf{x}_i)]^2 \quad (9)$$

Assuming the basis functions in $\boldsymbol{\phi}_{n+1}$ are all known, then according to (5)

$$\mathbf{w}_{n+1} = \mathbf{M}_{n+1}^{-1}\{\beta_i\boldsymbol{\phi}_{n+1,i}y_i\}_i \quad (10)$$

minimizes (9), where the Fisher information matrix $\mathbf{M}_{n+1}$ is given as

$$\mathbf{M}_{n+1} = \{\beta_i\boldsymbol{\phi}_{n+1,i}\boldsymbol{\phi}_{n+1,i}^T\}_i \quad (11)$$

It can be shown [7] that $\mathbf{w}_{n+1}$ is related to $\mathbf{w}_n$ as

$$\mathbf{w}_{n+1} = \begin{bmatrix} \mathbf{w}_n + \mathbf{M}_n^{-1}\{\beta_i\boldsymbol{\phi}_{n,i}\phi_{n+1,i}\}_i b^{-1}[\{\beta_i\boldsymbol{\phi}_{n,i}^T\phi_{n+1,i}\}_i\mathbf{w}_n - \{\beta_i\phi_{n+1,i}y_i\}_i] \\ -b^{-1}\{\beta_i\boldsymbol{\phi}_{n,i}^T\phi_{n+1,i}\}_i\mathbf{w}_n + b^{-1}\{\beta_i\phi_{n+1,i}y_i\}_i \end{bmatrix} \quad (12)$$

and e_{n+1} is related to e_n as

$$e_{n+1} = e_n - \delta e(K,\mathbf{c}_{n+1}) \quad (13)$$

where in (12) and (13),

$$\delta e(K,\mathbf{c}_{n+1}) = (1/\sum_{i=1}^{N}\beta_i)b^{-1}[\{\beta_i\boldsymbol{\phi}_{n,i}^T\phi_{n+1,i}\}_i\mathbf{w}_n - \{\beta_i\phi_{n+1,i}y_i\}_i]^2 \quad (14)$$

and

$$b = \{\beta_i\phi_{n+1,i}^2\}_i - \{\beta_i\phi_{n+1,i}\boldsymbol{\phi}_{n,i}^T\}_i\mathbf{M}_n^{-1}\{\beta_i\boldsymbol{\phi}_{n,i}\phi_{n+1,i}\}_i \quad (15)$$

with $\phi_{n+1,i} = K(\mathbf{c}_{n+1},\mathbf{x}_i)$.

It can be shown that b^{-1} is a diagonal element of $\mathbf{M}_{n+1}^{-1}$ [7]. With sufficient independent training data, we can always make $\mathbf{M}_{n+1}$ positive definite, as can be seen from (11). Then $\mathbf{M}_{n+1}^{-1}$ is also positive definite and it holds $b^{-1} > 0$, and therefore $\delta e(K,\mathbf{c}_{n+1})$ is non-negative. Then according to (13), $e_{n+1} < e_n$, which means appending a new basis function to the KMP generally leads to decrease of the representation error.

As $\delta e(K,\mathbf{c}_{n+1})$ is dependent on the center $\mathbf{c}_{n+1}$ of the new basis function, we obtain different values of $\delta e(K,\mathbf{c}_{n+1})$ by selecting different $\mathbf{c}_{n+1}$. If we confine $\mathbf{c}_{n+1}$ to be selected from the training data, we then can conduct a "greedy" search in the training set but with the previously selected data excluded to avoid repetition, and select the datum that maximizes (14). Formally, we have

$$\mathbf{c}_{n+1} = \mathbf{x}_{i_{n+1}} = \arg\max_{\substack{k\neq i_1,...,i_n \\ 1\leq k\leq N}} \delta e(K,\mathbf{x}_k) \quad (16)$$

After $\mathbf{c}_{n+1}$ is determined, we update the weights using (12) and the Fisher information matrix using (11) and (8).

2.1.3 Kernel Optimization

From (14), $\delta e(K,\mathbf{c}_{n+1})$ depends on the functional form of the kernel $K(\cdot,\cdot)$ as well as on $\mathbf{c}_{n+1}$. This allows us to optimize the kernel to gain further error reduction. A simple approach to take is to first conduct a "greedy" search of $\mathbf{c}_{n+1}$ in the training set, for a fixed kernel, and then fix $\mathbf{c}_{n+1}$ and optimize the parameters of the kernel. For radial basis function (RBF) kernels, the only parameter other than $\mathbf{c}_{n+1}$ is the kernel width, thus optimization of RBF kernels with $\mathbf{c}_{n+1}$ fixed is a one-dimensional search for the kernel width. It is also possible to optimize $\mathbf{c}_{n+1}$ and the kernel width simultaneously, but then $\mathbf{c}_{n+1}$ is treated as a free parameter and no longer confined to the training set. Another possibility is optimization over kernels of different functional forms, which offers greater diversity of the basis functions available to the KMP.

2.2 An M-ary KMP Classifier

For the M-class classification problem, one builds M models defined in (1). Suppose the training samples are $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ where $\mathbf{x}_i$ is an observed datum and $y_i \in \{1,2,...,M\}$ is its target label. One re-labels the training data for each of the M models in the following way. Let the labels for the m-th model be denoted as $y_i^{(m)}$, $i = 1,2,...,N$, then

$$y_i^{(m)} = \begin{cases} 1, & \text{if } y_i = m \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

The learning is based on *simultaneous* minimization of the weighted sum of squared errors for the M models. Thus the cost function for the M-ary KMP classifier is

$$e_n = (1/\sum_{i=1}^{N}\beta_i)\sum_{m=1}^{M}\sum_{i=1}^{N}\beta_i[y_i^{(m)} - \boldsymbol{\phi}_n^T(\mathbf{x}_i)\mathbf{w}_n^{(m)}]^2 \quad (18)$$

Note in (18) that the M models have their own weights but share the same basis functions. As in the case of the KMP, we first solve for the weights assuming the basis functions (including kernel parameters) are fixed. This is done by taking derivative of (18) with respect to $\mathbf{w}_n^{(m)}$, setting the result to zero, and solving for $\mathbf{w}_n^{(m)}$

$$\mathbf{w}_n^{(m)} = \mathbf{M}_n^{-1}\{\boldsymbol{\phi}_{n,i}y_i^{(m)}\}_i \quad (19)$$

where $\mathbf{M}_n$ is the same as in (6). Following the same methods that were used to derive (12)-(13), we obtain

$$\mathbf{w}_{n+1}^{(m)} = \begin{bmatrix} \mathbf{w}_n^{(m)} + \mathbf{M}_n^{-1}\{\beta_i\boldsymbol{\phi}_{n,i}\phi_{n+1,i}\}_i b^{-1}[\{\beta_i\boldsymbol{\phi}_{n,i}^T\phi_{n+1,i}\}_i\mathbf{w}_n^{(m)} - \{\beta_i\phi_{n+1,i}y_i^{(m)}\}_i] \\ -b^{-1}\{\beta_i\boldsymbol{\phi}_{n,i}^T\phi_{n+1,i}\}_i\mathbf{w}_n^{(m)} + b^{-1}\{\beta_i\phi_{n+1,i}y_i^{(m)}\}_i \end{bmatrix} \quad (20)$$

and

$$e_{n+1} = e_n - \delta e(K,\mathbf{c}_{n+1}) \quad (21)$$

where

$$\delta e(K,\mathbf{c}_{n+1}) = (1/\sum_{i=1}^{N}\beta_i)b^{-1}\sum_{m=1}^{M}[\{\beta_i\boldsymbol{\phi}_{n,i}^T\phi_{n+1,i}\}_i\mathbf{w}_n^{(m)} - \{\beta_i\phi_{n+1,i}y_i^{(m)}\}_i]^2 \quad (22)$$

and b is the same as in (15). The learning of KMP classifiers proceeds in a similar generative way as described Sec. 2.1. At the n-th iteration, we first select $\mathbf{c}_{n+1}$ from the training data set (with the previously selected data excluded) that maximizes (22), to locate the new basis function, and then use (20) to update the weights. We can similarly optimize the kernels in the KMP M-ary classifier, using (22) as the objective function to optimize kernel parameters or select different kernel functional forms.

3. MULTI-ASPECT HRR TARGET CLASSIFICATION WITH THE M-ARY KMP CLASSIFIERS

We now consider high-range-resolution (HRR) radar classification of moving targets, for which the target's orientation relative to the sensor may be known approximately from the Doppler signatures. Two basic properties are known for the HRR waveforms [8]. First, they are not aligned temporally to one another, making inner product an inappropriate similarity measure. Secondly, the waveforms vary significantly, implying that the HRR waveforms are a strong function of the viewing angle. We assume a fixed depression angle and therefore variation is in the azimuthal angle ϕ only. To simplify the formulae, the 2π-modulus property of ϕ is ignored in the following, with the results readily modified to account for this.

To deal with the angular dependence of HRR waveforms on ϕ, the concept of target "states" was introduced in [9], where a state was defined as a contiguous range of angles for which the scattering physics is approximately stationary. Assume that a target is characterized by L states. The i-th state is specified by $q_i(\phi) \overset{Def.}{=} \Pr(s_i \mid \phi)$, which is the probability that the waveform $\mathbf{x}$ is in state s_i given the *exact* azimuth ϕ of $\mathbf{x}$, where Pr is an abbreviation of probability. We assume $q_i(\phi)$ is Gaussian with mean μ_i and variance η_i^2, for i=1,…,L. In reality, the exact azimuth ϕ cannot be known, and only an estimate $\hat{\phi}$ of ϕ can be obtained. Assume the error $e = \phi - \hat{\phi}$ is governed by a zero-mean Gaussian $p_e(\phi)$ with variance σ^2. Using the definitions of $q_i(\phi)$ and $p_e(\phi)$ and their Gaussianity assumptions, we have

$$\Pr(s_i \mid \hat{\phi}) = \int \Pr(s_i \mid \phi) \Pr(\phi \mid \hat{\phi}) d\phi = \int q_i(\phi) p_e(\hat{\phi} - \phi) d\phi \quad (23)$$

Moreover, $\Pr(s_i \mid \hat{\phi})$ is also a Gaussian distribution in $\hat{\phi}$ with mean μ_i and variance $\eta_i^2 + \sigma^2$. The $\Pr(s_i \mid \hat{\phi})$ defines the probability that waveform $\mathbf{x}$ is in state s_i given the azimuth *estimate* $\hat{\phi}$ of $\mathbf{x}$.

The angular dependence aforementioned can be employed to construct sequences of HRR waveforms, to enhance the classification performance. Let $\mathbf{x}_j$ be the jth HRR waveform in a length-J sequence, and let the estimated azimuth of $\mathbf{x}_j$ be $\hat{\phi}_j$. Our objective is to perform target classification based on the sequence $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_J\}$ and the associated azimuth estimates $\{\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_J\}$.

Assume there are a total of M targets in consideration. The probability of target T_m given the sequence $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_J\}$ and the associated azimuth estimates $\{\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_J\}$ is represented as

$$\Pr(T_m \mid \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_J, \hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_J) = \prod_{j=1}^{J} \sum_{i=1}^{L} \Pr(T_m \mid s_i, \mathbf{x}_j) \Pr(s_i \mid \hat{\phi}_j) \quad (24)$$

where $\Pr(s_i \mid \hat{\phi}_j)$ are the same as in (23) expect that they are normalized such that $\sum_i^L \Pr(s_i \mid \hat{\phi}_j) = 1$.

We assume the definitions of $q_i(\phi)$, i=1,…,L, and $p_e(\phi)$ are constant across all the M targets in consideration. Given that we are in a particular state s_i and that $\mathbf{x}_j$ is observed, we implement $\Pr(T_m \mid s_i, \mathbf{x}_j)$ using either the RVM [6] or the M-ary KMP classifier. For state s_i and target T_m, the RVM and KMP outputs both take the form

$$f_i^{(m)}(\mathbf{x}) = \sum_{l=1}^{N_i} w_{l,i}^{(m)} K(\mathbf{c}_l, \mathbf{x}) + w_{0,i}^{(m)} \quad (25)$$

where $\{\mathbf{c}_l\}_{l=1}^{N_i}$ are the training examples of all M targets in state s_i. As noted in (23), state s_i is defined in a "soft" way by a probability distribution $\Pr(s_i \mid \hat{\phi})$ in $\hat{\phi}$. To obtain $\{\mathbf{c}_l\}_{l=1}^{N_i}$, $\Pr(s_i \mid \hat{\phi})$ is truncated to produce "hard" boundaries between states. In (25) the RVM and the KMP are distinguished in the following way: the RVM starts with non-zero weights on all N_i training examples, and then iteratively sets most of the weights to zero, using a sparseness prior on the weights. The KMP starts with all N_i weights zero (no basis functions) and iteratively and sequentially adds basis functions.

Though "hard" state boundaries are required by the RVM, they are not necessary for the KMP, as for the KMP, $\Pr(s_i \mid \hat{\phi})$ in (23) afford the weights β in (4) and (18) and therefore the overlap between states can be handled using importance weighting. While this may yield more accurate classification, it is more time consuming as it takes more training examples for each state. We did not implement it in our present results due to the computational cost.

The RVM introduces a link function, which is used in both the training and testing phase. In the testing phase the RVM computes the probability of associating $\mathbf{x}$ with state s_i of target T_m as [6]

$$\Pr(T_m \mid s_i, \mathbf{x}) = [1 + \exp(-f_i^{(m)}(\mathbf{x}))]^{-1} \quad (26)$$

For the RVM design, the expression in (26) is used to represent $\Pr(T_m \mid s_i, \mathbf{x}_j)$ in (24). The KMP does not employ a link function in its training phase. However, in the testing phase a monotonic nonlinear transform is employed in the KMP to map its output to within the range of probability [0,1]. Based on the re-labeling scheme in (17), we here let the mapping take the form $\Pr(T_m \mid s_i, \mathbf{x}) = [1 + \exp(2 f_i^{(m)}(\mathbf{x}) - 1)]^{-1}$. In both the RVM and the KMP, $\Pr(T_m \mid s_i, \mathbf{x})$ is normalized such that $\sum_{i=1}^{L} \Pr(T_m \mid s_i, \mathbf{x}) = 1$.

An advantage of the RVM and the KMP is that they are applicable to arbitrary kernels or basis functions. We exploit this property in this HRR classification problem. Let $C(\mathbf{x}_i, \mathbf{x}_j)$ be the *maximum* correlation between HRR waveforms $\mathbf{x}_i$ and $\mathbf{x}_j$, with the maximization performed across all possible temporal shifts between these waveforms. We define the kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-\frac{1}{2\gamma^2}[C(\mathbf{x}_i, \mathbf{x}_i) + C(\mathbf{x}_j, \mathbf{x}_j) - 2C(\mathbf{x}_i, \mathbf{x}_j)]\} \quad (27)$$

which is of the form of the radial basis function [3] except that the maximum correlation between $\mathbf{x}_i$ and $\mathbf{x}_j$ is used in place of the inner product. Note if the HRR waveforms are normalized in advance such that $C(\mathbf{x}, \mathbf{x}) = 1$, computational benefits are gained. The $K(.,.)$ in (27) does not appear to be a Mercer kernel, making it unsuitable for the support vector machine (SVM).

4. RESULTS

The multi-aspect processing approach in Sec. 3 has been developed for moving targets, for which the approximate azimuth estimates are available from the Doppler information. The assumption of available estimates for the target pose is most appropriate for sensing airborne targets, for which Doppler information is readily available. For presentation purposes, we present example results using the publicly available MSTAR data set [10]. The MSTAR data was originally in the form of synthetic-aperture radar (SAR) imagery, and it has recently been converted to HRR time-domain waveforms, as a function of azimuth angles [10]. In this data set, the training and testing data are distinct.

In the training phase, L=120 uniform target states were considered over the 360° azimuthal range. Accordingly the $\Pr(s_i \mid \hat{\phi})$ in (23) is a Gaussian with mean μ_i=1.5°+(i–1)×3° and variance $\eta_i^2+\sigma^2$=2.25°. The small variance used here is due to the fact that the azimuths furnished in the MSTAR dataset are very accurate (therefore σ^2 is small) and that the HRR waveforms vary fast with azimuth for the small wavelength here (3 cm) relative to the targets' dimensions (therefore η_i^2 with i=1,…,L are small). In training the RVM or KMP of the form in (25), $\Pr(s_i \mid \hat{\phi})$ is truncated to produce uniform states of 3° azimuthal support. The data were sampled in 0.1° increments azimuthally, and therefore for a given target each state was composed of 30 training examples. Across the M=10 targets the total number of training examples for a given state was 300.

We train an M-ary classifier for each of the 120 target states, based on the re-labeling in (17), and respectively using the RVM an KMP. The γ in (27) is initially chosen as 0.7. For the KMP γ is optimized on each iteration when learning, in the sense discussed in Sec. 2.1.3, while it is kept constant for the RVM. Recall that the RVM requires inversion of matrices of the size of the training set, and therefore the computation is often intensive. In the present example, the RVM required a total training time of 3.5 hours on a Pentium IV PC with 1.5 GHz clock speed. By comparison, the KMP training avoids large matrix inversions, and therefore a total training time of only 30 minutes is required on the same computer used for the RVM.

In Tables 1 and 2 we present the confusion matrices for the ten MSTAR targets, using the algorithms presented above. In Table 1 the confusion matrix is presented when $\Pr(T_m | s_i, \mathbf{x}_k)$ is modeled via the RVM. This table was generated by considering all possible 3° testing sequences, corresponding to 30 HRR waveforms collected in contiguous 0.1° increments. In this example 52 % of the training data were used as relevant vectors. The corresponding KMP results are presented in Table 2, for which 22% of the training data were used as bases. We observe that the two approaches yield comparable and encouraging performances, with an average classification rate of 96.6% for the RVM and 97.6% for the KMP.

5. CONCLUSIONS

KMP has been re-formulated in the framework of the theory of optimal experiments [5] and extended to M-ary classification and kernel optimization, using a weighted sum of squared errors as the loss function. The M-ary KMP classifier has been used as a key component in a multi-aspect classification scheme, to handle the data of M targets in a same state. The results on ten MSTAR targets show that at a comparable classification rate, the KMP achieves greater sparsity and shorter training time than the RVM.

6. ACKNOWLEDGMENT

This work has been carried out under the supervision of Prof. Lawrence Carin of Duke University.

7. REFERENCES

[1] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397–3415,Dec. 1993.

[2] P. Vincent and Y. Bengio. "Kernel matching pursuit", *Machine Learning*, 48, 165-187, 2002.

[3] S. Chen, F. Cowan, and P. Grant, "Orthogonal least squares learning algorithm for radial basis function networks", *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, 302-309, 1991

[4] http://www.anc.ed.ac.uk/~mjo/intro/node1.html

[5] V. V. Fedorov, *Theory of Optimal Experiments*, Academic Press, 1972

[6] M. Tipping, "Sparse Bayesian learning and the relevance vector machine", *Journal of Machine Learning Research*, 1, 2001, pp. 211-244.

[7] X. Liao, H. Li, B. Krishnapuram, and L. Carin, "A Generative Learning Method for Kernel Machines with Application to Multi-aspect Target Identification", Technical Report, February, 2003, Department of ECE, Duke University

[8] S. Hudson and D. Psaltis, "Correlation filters for aircraft identification from radar range profiles", *IEEE Trans. Aero. and Electronics System*, Vol. 29, No.3, pp. 741-748, 1993.

[9] P. Runkle, P. Bharadwaj, and L. Carin, "Hidden Markov models for multi-aspect target classification," *IEEE Trans. Signal Proc.,* vol. 47, pp. 2035-2040, July 1999.

[10] X. Liao, P. Runkle, L. Carin, "Identification of ground targets from sequential HRR radar signatures", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 4, Oct 2002, pp.1230 –1242.

Table 1. Confusion matrix of RVM, 52% of the training data used as relevant vectors. Each test sequence spans 3° in azimuth. The (i,j) element of the matrix is the rate in percentage that the test sequences of target T_i are declared as from target T_j.

	T72	BTR70	BMP2	2S1	ZSU234	BTR60	BRDM2	D7	T62	ZIL131
T72	98.29	0.00	0.00	0.00	0.00	0.34	0.00	0.00	0.00	0.00
BTR70	0.00	94.04	0.56	0.50	0.00	0.03	2.63	0.00	0.00	0.00
BMP2	0.00	0.17	98.77	0.39	0.08	0.00	0.42	0.06	0.00	0.11
2S1	0.00	0.31	0.98	97.79	0.00	0.20	0.70	0.03	0.00	0.00
ZSU234	1.12	0.28	0.20	0.64	96.25	0.08	0.28	0.00	0.64	0.50
BTR60	0.00	2.18	0.50	0.00	0.00	95.88	0.81	0.00	0.00	0.00
BRDM2	0.00	0.00	0.22	0.00	0.00	0.00	99.30	0.00	0.00	0.00
D7	0.03	0.00	0.00	0.08	0.00	0.00	0.00	99.30	0.45	0.14
T62	0.92	0.00	0.20	0.92	0.70	0.08	0.06	0.28	95.04	0.42
ZIL131	0.00	1.04	0.76	0.87	2.13	0.42	0.87	0.42	1.54	91.83
average	96.65									

Table 2. Confusion matrix of KMP, 22% of the training data used as kernel centers. Each test sequence spans 3° in azimuth.

	T72	BTR70	BMP2	2S1	ZSU234	BTR60	BRDM2	D7	T62	ZIL131
T72	99.66	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.03	0.25
BTR70	0.00	94.37	0.42	0.78	0.00	0.67	2.77	0.00	0.00	0.98
BMP2	0.48	0.06	98.91	0.00	0.00	0.56	0.00	0.00	0.00	0.00
2S1	0.53	0.03	0.53	98.18	0.03	0.00	0.70	0.00	0.00	0.00
ZSU234	0.25	0.64	0.11	0.00	96.98	0.42	0.00	0.70	0.81	0.08
BTR60	0.00	1.65	0.17	0.00	0.00	96.19	0.06	0.03	0.34	1.57
BRDM2	0.00	0.00	0.00	0.42	0.08	0.22	99.27	0.00	0.00	0.00
D7	0.06	0.00	0.00	0.00	0.00	0.00	0.00	99.92	0.03	0.00
T62	0.22	0.03	0.73	0.00	0.25	1.06	0.00	0.00	97.12	0.59
ZIL131	0.00	1.96	0.03	0.14	0.28	0.00	1.65	0.00	0.00	95.94
average	97.65									