Beamforming Using Support Vector Machines

M. Martínez Ramón, Nan Xu, and C. G. Christodoulou, *Fellow, IEEE*

***Abstract*—Support vector machines (SVMs) have improved generalization performance over other classical optimization techniques. Here, we introduce an SVM-based approach for linear array processing and beamforming. The development of a modified cost function is presented and it is shown how it can be applied to the problem of linear beamforming. Finally, comparison examples are included to show the validity of the new minimization approach.**

***Index Terms*—Beamforming, complex support vector machines, support vector machines (SVMs).**

I. Introduction

SUPPORT vector machines (SVMs) have shown several advantages in prediction, regression, and estimation over some of the classical approaches in a wide range of applications due to its improved generalization capabilities. Here, we introduce the basic framework of the SVM approach as applied to linear array processing.

Array signal processing involves complex signals, for which a complex-valued formulation of the SVM is needed. We introduce this formulation by introducing the real and imaginary parts of the error in the primal optimization and then proceeding as usual to solve a complex valued constrained optimization problem. The resulting algorithm is a natural counterpart of the real valued support vector regressor, which can be immediately applied to array signal processing. The adjustment of the parameters into this cost function leads to an improved robustness of the method in the presence of any additional noise in the signal.

We apply the newly developed formulation in optimizing the beamforming from an array antenna of six elements as a proof of concept. Several examples illustrate the advantage of SVMs over minimum mean square error (MMSE)-based algorithms due to its improved generalization ability. The first examples compare the behavior of both algorithms in an environment in which interfering signals are close to the desired ones, thus producing non-Gaussian noise. The last example illustrates the improved generalization ability of the SVM when small data sets are used for training, which is common in several communication applications.

II. The Support Vector Approach and the Cost Function

Let $x[n]$ be spatially sampled data. A linear beamformer can be written as

$$d[n] = \mathbf{w}^T\mathbf{x}[n] + \epsilon[n] \tag{1}$$

where $\mathbf{x}[n]$ is the vector of M elements of the array output and $\epsilon[n]$ is the output error.

Coefficients $\mathbf{w}$ are usually estimated through the minimization of a certain cost function on $\epsilon[n]$.

The SVM approach can be applied to the adjustment of this model. The main idea of SVMs is to obtain the solution which has the minimum norm of $\mathbf{w}$. Due to the minimization of the weight vector norm, the solution will be regularized in the sense of Thikonov [1]), improving the generalization performance. The minimization has to be subject to the constraints

$$\begin{aligned} d[n] - \mathbf{w}^T\mathbf{x}[n] &\le \varepsilon + \xi_n \\ -d[n] + \mathbf{w}^T\mathbf{x}[n] &\le \varepsilon + \xi_n' \\ \xi[n], \xi'[n] &\ge 0 \end{aligned} \tag{2}$$

not to be trivial. ξ_n and ξ_n' are the "slack" variables or losses. The optimization is intended to minimize a cost function over these variables. The parameter ε is used to allow those ξ_n or ξ_n' for which the error is less that ε to be zero. This is equivalent to the minimization of the so-called ε-insensitive or Vapnik loss function [2], given by

$$L_\varepsilon(\epsilon) = \begin{cases} |\epsilon| - \varepsilon, & |e| > \varepsilon \\ 0, & |e| < \varepsilon \end{cases} \tag{3}$$

The functional to be minimized is then

$$L_p = ||\mathbf{w}||^2 + C\sum_n L_\varepsilon\left(\xi_n, \xi_n'\right) \tag{4}$$

subject to $\xi_n, \xi_n' \ge 0$ where C is the tradeoff between the minimization of the norm (to improve generalization ability) and the minimization of the errors [2].

The optimization of the above constrained problem through Lagrange multipliers α_i, α_i' leads to the dual formulation [3]

$$L_d = -(\boldsymbol{\alpha} - \boldsymbol{\alpha}')^T\mathbf{R}(\boldsymbol{\alpha} - \boldsymbol{\alpha}') + (\boldsymbol{\alpha} - \boldsymbol{\alpha}')^T\mathbf{y} - (\boldsymbol{\alpha} + \boldsymbol{\alpha}')\mathbf{1}\varepsilon \tag{5}$$

to be minimized with respect to $(\alpha_i - \alpha_i')$.

It involves the Gram matrix $\mathbf{R}$ of the dot products of the data vectors $\mathbf{x}[n]$. This matrix may be singular and thus producing an ill-conditioned problem. To avoid this numerical inconvenience, a small diagonal $\gamma\mathbf{I}$ is added to the matrix prior to the numerical optimization.

We present here a modified derivation of the SVM regressor which leads to a more convenient equivalent cost function (Fig. 1)

$$L_R(e) = \begin{cases} 0, & |e| < \varepsilon \\ \frac{1}{2\gamma}\left(|e| - \varepsilon\right)^2 - \varepsilon, & \varepsilon \le |e| \le \varepsilon + e_C \\ C\left(|e| - \varepsilon\right) - \frac{1}{2}\gamma C^2, & e_C \le |e| \end{cases} \tag{6}$$

Manuscript received July 14, 2005; revised September 27, 2005.

The authors are with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87106 USA.

Digital Object Identifier 10.1109/LAWP.2005.860196

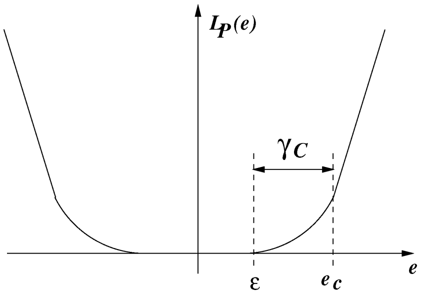

Fig. 1. Cost function applied to the SVM beamformer.

where $e_C = \varepsilon + \gamma C$.

This cost function provides a functional that is numerically regularized by the matrix γI. As it can be seen, the cost function is quadratic for the data which produce errors between ε and e_C, and linear for errors above e_C. Thus, one can adjust the parameter e_C to apply a quadratic cost for the samples which are mainly affected by thermal noise (i.e., for which the quadratic cost is maximum likelihood). The linear cost is then applied to the samples which are outliers [4], [5]. Using a linear cost function, the contribution of the outliers to the solution will not depend on its error value, but only on its sign, thus avoiding the bias that a quadratic cost function produces.

Finally, we generalize the derivation to the complex-valued case, as it is necessary for array processing.

III. Support Vector Machine Beamformer

The output vector of an M-element array receiving K signals can be written in matrix notation as

$$\mathbf{x}[\mathbf{n}] = \mathbf{A}\mathbf{s}[n] + \mathbf{g}[n] \tag{7}$$

where

$$\begin{aligned}
\mathbf{A} &= [\mathbf{a}(\theta_1) \cdots \mathbf{a}(\theta_k)] \\
\mathbf{a}(\theta_i) &= \left[1 e^{-jk_i} \cdots e^{-j(M-1)k_i}\right]^T \\
\mathbf{s}[n] &= [s_1[n] \cdots s_k[n]]^T \\
\mathbf{g}[n] &= [g_1[n] \cdots g_M[n]]^T
\end{aligned} \tag{8}$$

are respectively the $M \times K$ steering matrix and vectors, the received signals, and the noise present at the output of each array element.

The output vector $\mathbf{x}[\mathbf{n}]$ is linearly processed to obtain the desired output d[n]. The expression for the output of the array processor is

$$y[n] = \mathbf{w}^T \mathbf{x}[n] = d[n] + \epsilon[n] \tag{9}$$

where $\mathbf{w} = [w_1 \cdots w_M]$ is the weight vector of the array and $\epsilon[n]$ is the estimation error.

For a set of N observed samples of $\{x[n]\}$ and when nonzero empirical errors are expected, the functional to be minimized is:

$$\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} L_R\left(e[n], \varepsilon, \gamma, C\right). \tag{10}$$

Thus, according to the error cost function (6), we have to minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + \sum L_R\left(\xi_n + \xi'_n\right) + \sum L_R\left(\zeta_n + \zeta'_n\right) \tag{11}$$

subject to

$$\begin{aligned}
Re\left(d[n] - \mathbf{w}^T\mathbf{x}[n]\right) &\leq \varepsilon + \xi_n \\
Re\left(-d[n] + \mathbf{w}^T\mathbf{x}[n]\right) &\leq \varepsilon + \xi'_n \\
Im\left(d[n] - \mathbf{w}^T\mathbf{x}[n]\right) &\leq \varepsilon + \zeta_n \\
Im\left(-d[n] + \mathbf{w}^T\mathbf{x}[n]\right) &\leq \varepsilon + \zeta'[-] \\
\xi[n], \xi'[n], \zeta[n], \zeta'[n] &\geq 0
\end{aligned} \tag{12}$$

where $\xi[n]$, and $\xi'[n]$ stand for positive, and negative errors in the real part of the output, respectively. $\zeta[n]$ and $\zeta'[n]$ represent the errors for the imaginary part. Note that errors are either negative or positive and, therefore, only one of the losses takes a nonzero value, that is, either $\xi[n]$ or $\xi'[n]$ (either $\zeta[n]$ or $\zeta'[n]$) is null. This constraint can be written as $\xi[n]\xi'[n] = 0$ $(\zeta[n]\zeta'[n] = 0)$. Finally, as in other SVM formulations, the parameter C can be seen as a tradeoff factor between the empirical risk and the structural risk.

It is possible to transform the minimization of the primal functional (11) subject to constraints in (12), into the optimization of the dual functional or Lagrange functional. First, we introduce the constraints into the primal functional by means of Lagrange multipliers, obtaining the following primal-dual functional:

$$\begin{aligned}
L_{pd} = \frac{1}{2}\|\mathbf{w}\|^2 &+ C\sum_{n\in I_1}^{N}\left(\xi_n + \xi'_n\right) + C\sum_{n\in I_1}^{N}\left(\zeta_n + \zeta'_n\right) \\
&\times \frac{1}{2\gamma}\sum_{n\in I_2}^{N}\left(\xi_n^2 + \xi_n'^2\right) + \frac{1}{2\gamma}\sum_{n\in I_2}^{N}\left(\zeta_n^2 + \zeta_n'^2\right) \\
&- \sum_{n=n_0}^{N}\left(\lambda_n\xi_n + \lambda'_n\xi'_n\right) - \sum_{n=k_0}^{N}\left(\eta_n\zeta_n + \eta'_n\zeta'_n\right) \\
&\times \sum_{n=n_0}^{N}\alpha_n\left[Re\left(d[n] - \mathbf{w}^T\mathbf{x}[n]\right) - \varepsilon - \xi_n\right. \\
&\times \sum_{n=n_0}^{N}\alpha'_n\left[Re\left(-d[n] + \mathbf{w}^T\mathbf{x}[n]\right) - \varepsilon - \xi'_n\right. \\
&\times \sum_{n=n_0}^{N}\beta_n\left[Im\left(d[n] - \mathbf{w}^T\mathbf{x}[n]\right) - j\varepsilon - j\zeta_n\right. \\
&\times \sum_{n=n_0}^{N}\beta'_n\left[Im\left(-d[n] + \mathbf{w}^T\mathbf{x}[n] +\right) - j\varepsilon - j\zeta'_n\right]
\end{aligned} \tag{13}$$

with the dual variables or Lagrange multipliers constrained to $\alpha_n, \beta_n, \lambda_n, \eta_n, \alpha'_n, \beta'_n, \lambda'_n, \eta'_n \geq 0$ and with $\xi_n, \zeta_n, \xi'_n, \zeta'_n \geq 0$. Note that cost function (1) has two active segments, a quadratic one and a linear one.

The following constraints must also be fulfilled:

$$\begin{aligned}
\alpha_n\alpha'_n &= 0 \\
\beta_n\beta'_n &= 0.
\end{aligned} \tag{14}$$

Besides, the Karush–Kuhn–Tucker (KKT) conditions [2] force $\lambda_n\xi_n = 0$, $\lambda'_n\xi'_n = 0$ and $\eta_n\zeta_n = 0$, $\eta'_n\zeta'_n = 0$. Functional (13) has to be minimized with respect to the primal variables and maximized with respect to the dual variables. By minimizing L_{pd} with respect to w_i we obtain an optimal solution for the weights

$$\mathbf{w} = \sum_{n=0}^{N} \psi_n \mathbf{x}^*[n] \tag{15}$$

where $\psi_n = \alpha_n - \alpha'_n + j(\beta_n - \beta'_n)$. This result is analogous to the one for the real-valued SVM problem, except that now Lagrange multipliers α_n and β_n for both real and imaginary components have been considered. Optimizing L_{pd} with respect to ξ_n and ζ_n and applying the KKT conditions leads to an analytical relationship between the residuals and the Lagrange multipliers. This relationship is given by

$$\begin{aligned}(\alpha - \alpha') &= \begin{cases} -C, & Re(e) \le -e_C \\ \frac{1}{\gamma}\left(Re(e) + \varepsilon\right), & -e_C \le Re(e) \le -\varepsilon \\ 0, & -\varepsilon \le Re(e) \le \varepsilon \\ \frac{1}{\gamma}\left(Re(e) - \varepsilon\right), & \varepsilon \le Re(e) \le e_C \\ C, & e_C \le Re(e) \end{cases} \\ (\beta - \beta') &= \begin{cases} -C, & Im(e) \le -e_C \\ \frac{1}{\gamma}\left(Im(e) + \varepsilon\right), & -e_C \le Im(e) \le -\varepsilon \\ 0, & -\varepsilon \le Im(e) \le \varepsilon \\ \frac{1}{\gamma}\left(Im(e) - \varepsilon\right), & \varepsilon \le Im(e) \le e_C \\ C, & e_C \le -Im(e). \end{cases}\end{aligned} \tag{16}$$

As mentioned earlier, it is possible to continue toward the purely dual formulation of the problem that can be solved using quadratic programming (QP) as usually is done in the literature but, this approach is computationally intensive, and lacks the required flexibility for our communications problem at hand. Alternative optimization methods such as those relying on the iterative reweighted least squares (IWRLS) have been introduced in [6] with a clear advantages in terms of computational cost and flexibility of operation.

Using (15), the norm of the complex coefficients can be written as

$$||\mathbf{w}||^2 = \sum_{i=0}^{N} \sum_{j=n_0}^{N} \psi_j \psi_i^* \mathbf{x}[j]\mathbf{x}^*[i]. \tag{17}$$

By using matrix notation again and storing all partial correlations in (17), we can write

$$R[j,i] = \mathbf{x}[j]\mathbf{x}^*[i] \tag{18}$$

so that the norm of the coefficients can be written

$$||\mathbf{w}||^2 = \boldsymbol{\psi}^H \mathbf{R} \boldsymbol{\psi} \tag{19}$$

$\mathbf{R}$ being the matrix with elements $R[j,i]$, and $\psi = (\psi_{n_0} \ldots \psi_N)^T$. By substituting (15) in functional (13), the dual functional to be maximized is as follows:

$$\begin{aligned} L_d = &\frac{1}{2}\boldsymbol{\psi}^H\mathbf{R}\boldsymbol{\psi} - Re\left[\boldsymbol{\psi}^H\mathbf{R}(\boldsymbol{\alpha} - \boldsymbol{\alpha}')\right] \\ &+ Im\left[\boldsymbol{\psi}^H\mathbf{R}(\boldsymbol{\beta} - \boldsymbol{\beta}')\right] \\ &+ Re\left[(\boldsymbol{\alpha} - \boldsymbol{\alpha}')^T\mathbf{y}\right] - Im\left[(\boldsymbol{\beta} - \boldsymbol{\beta}')^T\mathbf{y}\right] \\ &- (\boldsymbol{\alpha} + \boldsymbol{\alpha}')\mathbf{1}\varepsilon - (\boldsymbol{\beta} + \boldsymbol{\beta}')\mathbf{1}\varepsilon + L_C \end{aligned} \tag{20}$$

L_C being a function of ψ.

Intervals I_1 and I_2 must be treated separately.

- Using (16) into interval I_1 yields $\alpha_m^{(\prime)} = \beta_m^{(\prime)} = C$. Then, the last term of the functional for I_1 becomes

$$L_C(I_1) = C\mathbf{I} \tag{21}$$

where $\mathbf{I}$ is the identity matrix.

- Using (16) into interval I_2, then $\alpha_m^{(\prime)} = (1/\gamma)\xi_m^{(\prime)}$ and $\beta_m^{(\prime)} = (1/\gamma)\zeta_m^{(\prime)}$. The last term for this interval becomes

$$L_C(I_2) = \frac{\gamma}{2}\psi_2^H \mathbf{I} \psi_2 \tag{22}$$

ψ_2 being the elements of interval I_2.

Both terms can be grouped as

$$L_C = \frac{\gamma}{2}\psi^H I \psi + \left(1 - \frac{\gamma}{2}\right) C D_{I_2} \tag{23}$$

D_{I_2} being a diagonal matrix with terms corresponding to I_1 set to 1 and the remaining set to 0. As the last term of L_C is just a constant, it can be removed from the optimization.

By regrouping terms and taking into account that $\psi^H\mathbf{R}\psi = \psi^H Re(\mathbf{R})\psi$, the functional (20) can be written in a more compact form as

$$L_d = -\frac{1}{2}\psi^H Re\left(\mathbf{R} + \frac{\gamma}{2}\mathbf{I}\right)\psi + Re[\psi^H\mathbf{y}] - (\alpha + \alpha' + \beta + \beta')\mathbf{1}\varepsilon. \tag{24}$$

IV. Examples

A. Bit Error Rate Performance

We first compared the algorithm with the regular least squares (LS) approach for an array of six elements. The desired signals come from the angles between -0.1π and 0.25π, with amplitudes 1 and 0.3, and the interfering signals come from within the range of -0.05π, 0.1π, and 0.3π with an amplitude of 1.

In order to train the beamformer, a burst of 50 known symbols is sent. Then, the bit error rate (BER) is measured with bursts of 10 000 unknown symbols.

In the first two examples, we fix γ of (6) to 10^{-6} and then the product γC is adjusted to the noise standard deviation. That way, most of the samples which are corrupted only by (thermal)

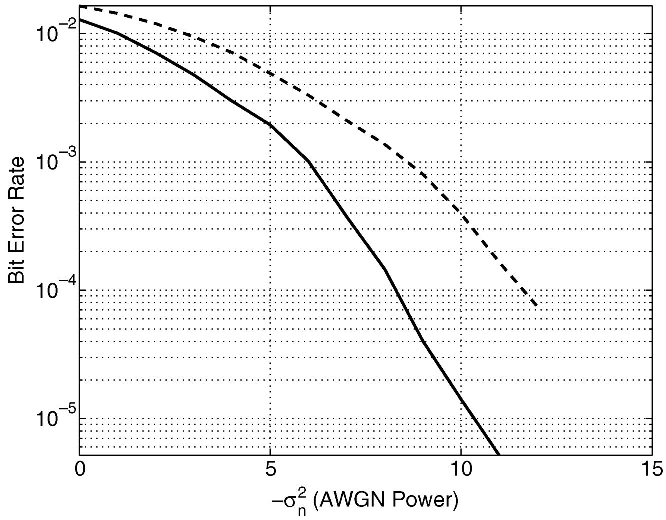

Fig. 2. BER performance for example 1. SVM (continuous line) and regular LS (dashed line) beamformers.

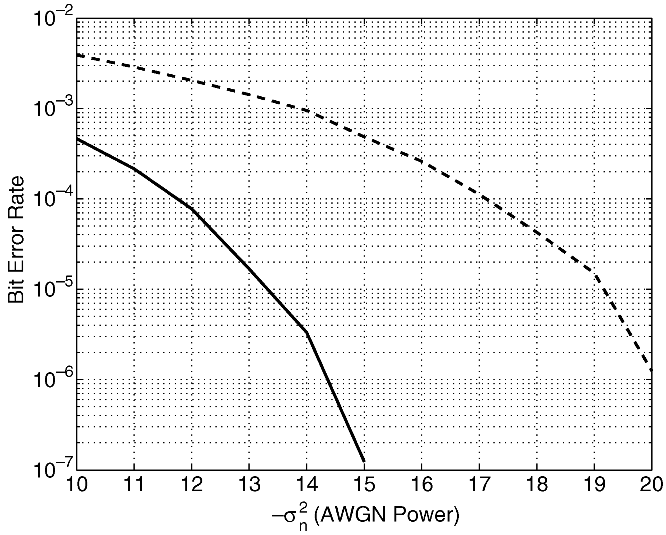

Fig. 3. BER performance for example 2. SVM (continuous line) and regular LS (dashed line) beamformers.

noise will fall in the quadratic area, whereas the outliers produced by interfering signals will fall in the linear area.

We calculated the BER performance of LS and SVM for different noise levels from 0 to -15 dB. Each BER has been evaluated by averaging the results of 100 independent trials. The results can be seen in Fig. 2.

In the next case, the desired signal coming from the angles of arrival (AOA) -0.1π and 0.25π, with amplitudes 1 and 0.3, where the interfering signals come from the AOAs -0.02π, 0.2π, 0.3π with amplitude 1 (see Fig. 3).

In the last example, the interfering signals are much closer to the desired ones, thus biasing the LS algorithm. The better performance of the SVM is due to its better robustness against the non-Gaussian outliers produced by the interfering signals.

B. Robustness Against Overfittng

One advantage of SVMs is that their generalization ability is controlled by the regularization imposed by the minimization of the weight vector norm. We highlight this fact by calculating the BER for different number of training samples. The results are shown in Fig. 4.

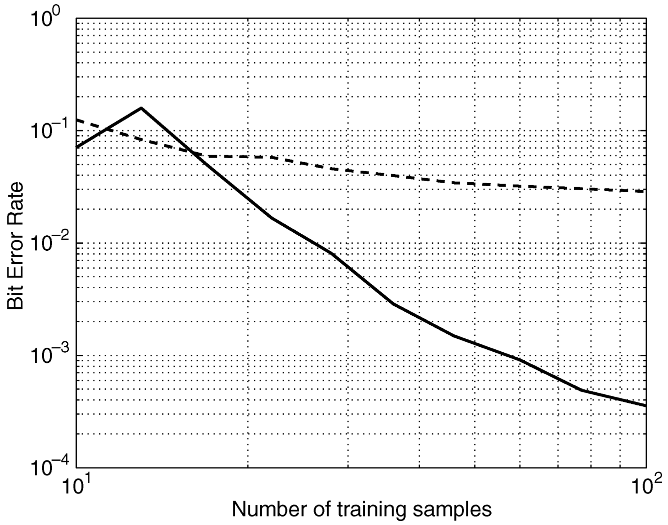

Fig. 4. BER performance against the number of training samples. SVM (continuous line) and regular LS (dashed line) beamformers.

V. Conclusion

We introduce in this work a way to apply the SVM framework to linear array beamforming. SVMs have a clear advantage over MMSE-based algorithms in those cases in which small data sets are available for training and where non-Gaussian noise is present, due to the fact that the generalization ability of the machine is controlled. In order to make the algorithm adequate to array processing purposes, we first apply an alternative cost function which is suitable in problems in which there are Gaussian noise and other non-Gaussian sources, as multiuser interference which may produce outliers in the signal. Also, this cost function provides a natural way to explain the numerical regularization present in any SVM.

Ongoing work is being done in the application of nonlinear SVMs for beamforming and detection of AOA.

References

[1] A. Tikhonov and V. Arsenen, *Solution to Ill-Posed Problems*. New York: Winston, 1977.

[2] V. Vapnik, *Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control*. New York: Wiley, 1998.

[3] A. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression," Royal Holloway College, Univ. London, U.K., NeuroCOLT Tech. Rep. NC-TR-98-030, 1988.

[4] P. J. Huber, "Robust statistics: a review," *Ann. Math. Statist.*, vol. 43, pp. 1041–1067, 1972.

[5] K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 243–254.

[6] A. Navia-Vázquez, F. Pérez-Cruz, A. Artés-Rodríguez, and A. Figueiras-Vidal, "Weighted least squares training of support vector classifiers leading to compact and adaptive schemes," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 1047–1059, Sep. 2001.