Support Vector Machines for Channel Equalization in Hostile Environments

Felix Albu
Faculty of Electronics & Telecommunications, 1-3 Bd. Iuliu Maniu, Bucharest,Romania
felix@comm.pub.ro , Tel: 401 410 54 00 Fax: 401 411 30 88

ABSTRACT **The paper investigates the application of Support Vector (SV) machines with Gaussian kernels for channel equalization, in the presence of intersymbol interference, additive white Gaussian noise and co-channel interference. We described the problems encountered by linear equalizers in these conditions. We made a comparison of the performances obtained for this task of the SV machines with a classical RBF network, where the centers are determined by *k*-means clustering, and the weights are found using the LMS algorithm. Our results show that the nonlinear equalizers offer a performance which exceeds that of linear structures. High generalization ability of SV machines for small training databases allows them to achieve better performance than the RBF equalizer, particularly for high SINRs.**

Index terms - Clustering, Co-channel interference, radial basis function networks, support vector machines

1. INTRODUCTION

Many communications systems are corrupted not only by channel intersymbol interference, but also by co-channel interference. The equalizers are required to obtain reliable data transmission. The most sophisticated equalizer structure is the maximum likelihood sequence estimator (MLSE), but is less effective in dealing with co-channel interference[1]. Most of the channel equalization applications employ a symbol-decision structure based on linear filtering. This is known as the linear transversal equalizer (LTE). Because the adaptive LTE converges to the Wiener filter, the performance of the Wiener filter forms the performance bound for the latter [2].

From the information contained in the equalizer inputs, processing methods yielding a better performance are employed. The equalization problem is treated as a classification task. It was shown that the radial basis function equalizer can exploit the difference between an interfering signal and the noise to the benefit of equalization performance[2]. In support vector classifiers, input vectors are mapped to a high dimensional space and are then separated by the optimal linear hyperplane. The traditional view of RBF networks has been one where the clustering heuristic used for training them was considered very important. In contrast, the SV algorithm provides a way of choosing the number and locations of RBF centers, with the centers being those examples that are critical for the classification task [3].

In the next sections we present the system model, the RBF equalizers, the Support Vector Machines (we refer mostly to the paper of D. Tax & all [4]), and some computer simulations showing the performance of these structures.

2. SYSTEM MODEL

The discrete-time model for the digital communications system considered here is depicted in Fig.1. In this model [5], $H_0(z)$ is the channel transfer function and there exists a total of p interfering co-channels with transfer functions $H_i(z), 1 \le i \le p$. The channels are usually modeled by an FIR filter with the following transfer function: $H_i(z) = \sum_{j=0}^{n_i} h_{ij} z^{-j}, 0 \le i \le p$, where h_{ij} are the channels impulse response components and n_i is their lengths. In our study the symbols $s_i(k)$ are taken from the data set {±1}; they form an i.i.d. sequences, and *e(k)* is an additive white Gaussian noise with zero mean and variance σ_e^2. The channel observation: $x(k) = \hat{y}(k) + u(k) + e(k)$ contains three terms called the desired signal, the interfering signal and the noise. Let $\mathrm{E}\left[\hat{y}^2(k)\right] = \sigma_{\hat{y}}^2$ and $\mathrm{E}\left[u^2(k)\right] = \sigma_u^2$. The signal to noise ratio is defined as $\mathrm{SNR} = \frac{\sigma_{\hat{y}}^2}{\sigma_e^2}$, the signal to interference ratio is given by $\mathrm{SIR} = \frac{\sigma_{\hat{y}}^2}{\sigma_u^2}$, and the signal to interference and noise ratio is $\mathrm{SINR} = \frac{\sigma_{\hat{y}}^2}{\sigma_e^2 + \sigma_u^2}$ [5].

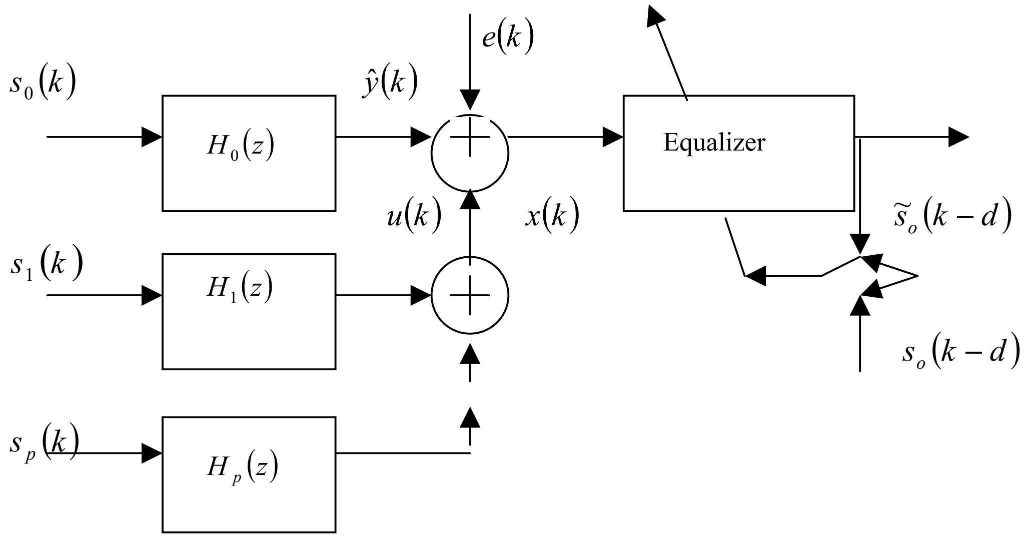


Fig.1 Discrete-time model of data transmission system

The task of the equalizer is to estimate the transmitted symbols $s_0(k)$ based on the channel observation $x(k)$. During the training period we know the reference signal $s_o(k-d)$, where the integer *d* reprezent the equalizer delay.
Let us consider the following co-channel system :

$$\begin{aligned} H_0(z) &= 0.5 + z^{-1} \\ H_1(z) &= 0.1 + 0.3z^{-1} \end{aligned} \tag{1}$$

In Fig. 2 we see the positions of desired signal states for the channel $H_0(z) = 0.5 + z^{-1}$ (denoted by o); the interfering channel $H_1(z) = 0.1 + 0.3z^{-1}$ causes an increase of the number of clusters signal states (denoted by *).

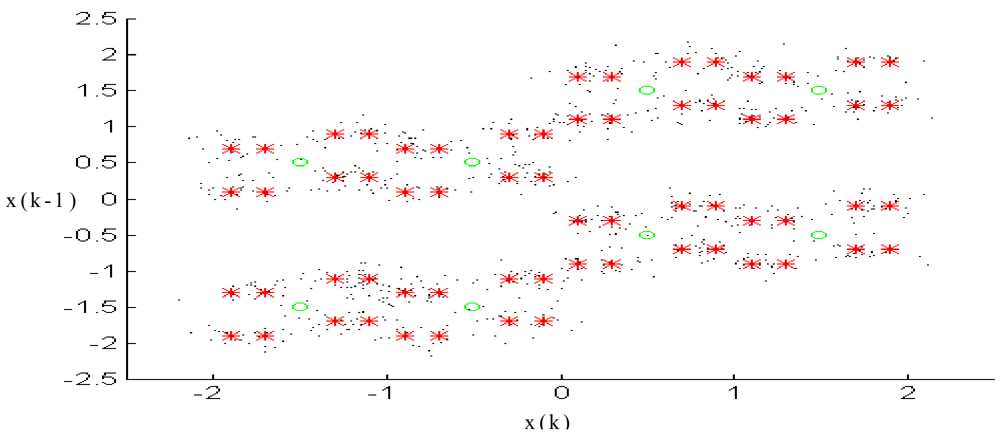


Fig. 2 Centers of clusters formed by $H_0(z) = 0.5 + z^{-1}$ (denoted by o), and by the interfering channel $H_1(z) = 0.1 + 0.3z^{-1}$ (denoted by *), 500 received data (SNR=20 dB)

The observations form clusters and the mean values of the data clusters are the noise-free observation states (see Fig. 2). If SIR is high, noise-free observation states will concentrate around the desired signal states. As SIR decreases, the noise-free observation states become more widely spread [5] .

3. RADIAL BASIS FUNCTIONS NETWORKS AS EQUALIZERS

A RBF network is a two-layer network comprising a hidden layer and an output layer. The hidden layer contains *n* neurons which compute the Euclidean distance between a center vector $\mathbf{c}_i$ and an input vector $\mathbf{x} = [x(t)\, x(t-1) \ldots x(t-m+1)]^t$. The result is passed through a nonlinear function Φ_i to generate the hidden node output. Functions Φ_i are chosen to be Gaussian: $\Phi_i = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c_i}\|^2}{\sigma_i^2}\right)$, where σ_i is called the width. The output layer is computed by a weighted linear combination of the *n* neurons of the hidden layer. The overall response is a mapping: $f\ (\mathbf{x}) = \sum_{i=1}^{n} w_i \Phi_i$, where w_i are the weights. It has been shown that RBFN perform an implementation of the optimal Bayesian equalizer if the channel is known and the parameters of the network are well chosen [2]. The RBF structure can be very complex for large m and n_h, but can be reduced however by some approximation , as shown in [5] or [6]. In our work, the training of RBFN was done using a two-steps approach: firstly a classical k-means clustering procedure was used to determine the location of the centers [7], and the widths were set as in [5]. Secondly, the weights were trained using the LMS algorithm. Whether a RBF network can realize optimal equalizer solution depend to a great extent on locating the centers at the desired channel states. If this is achieved, the full effects of the co-channel interference can be taken into account and the network can discriminate between the interfering signal and the noise [5]. This requires a great number of samples for the clustering procedure and a correct estimate of the channel order.

4. SUPPORT VECTOR MACHINES

4.1 THE OPTIMAL HYPERPLANE

The optimal hyperplane is defined as a plane separating two separable classes such that its margin is as wide as possible[3]. If we denote the training vectors by $\mathbf{x_i}, i = 1,\ldots,l$ with the corresponding labels $y_i \in \{-1,1\}$ then this yields

$$y_i[(\mathbf{w}\cdot\mathbf{x_i}) + b] \geq 1, \quad i = 1,\ldots,l \tag{2}$$

if the optimal hyperplane is

$$\mathbf{w}\cdot\mathbf{x} + b = 0 \tag{3}$$

in which **w** can be written as

$$\mathbf{w} = \sum_{\mathbf{i=1}}^{l} \alpha_i y_i \mathbf{x}_i, \quad \alpha_{\mathrm{i}} \geq 0 \tag{4}$$

The discrimination function for a certain vector **z** is therefore written as

$$f(z) = \sum_{i=1}^{l} \alpha_i y_i (\mathbf{z}\cdot\mathbf{x}_i) + b \tag{5}$$

It can be shown that in order to find the optimal set of weights α_i, the following expression has to be maximized:

$$W(\Lambda) = \Lambda^T \mathbf{I} - \frac{1}{2}\Lambda^{\mathbf{T}}\mathbf{D}\Lambda \tag{6}$$

w.r.t. $\Lambda = (\alpha_1,\ldots,\alpha_l)$, subject to the following constraints:

$$\begin{aligned} &\Lambda \geq 0 \\ &\mathbf{Y}^{\mathbf{T}}\Lambda = 0 \end{aligned} \tag{7}$$

D is a matrix containing dot products of the training vectors, multiplied by their labels:

$$\mathbf{D}_{ij} = y_i y_j (\mathbf{x}_i \mathbf{x}_j) \tag{8}$$

Maximizing Eq. (6) is a quadratic programming problem with linear constraints[4]. For the optimization problem we used an active set strategy (also known as a projection method) [8].

4.2 THE SOFT MARGIN HYPERPLANE

The concept of optimal hyperplane is only suitable for the separable case[4]. If classes overlap Eq. (2) is modified to allow for errors:

$$y_i\left[(\mathbf{w}\cdot\mathbf{x_i})+b\right]\geq 1-\xi_i,\quad i=1,\ldots,l$$
$$\xi_i\geq 0,\quad i=1,\ldots,l \tag{9}$$

and the following function can be minimized:

$$\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{l}\xi_i \tag{10}$$

The quadratic programming problem now becomes a dual one:

$$W(\Lambda,\delta)=\Lambda^T 1-\frac{1}{2}\left[\Lambda^{\mathbf{T}}\mathbf{D}\Lambda+\frac{\delta^2}{C}\right] \tag{11}$$

w.r.t. $\Lambda=(\alpha_1,\ldots,\alpha_l)$, and δ subject to constraints:

$$0\leq\Lambda\leq\delta\mathbf{1}$$
$$\mathbf{Y^T}\Lambda=0 \tag{12}$$

The dot product can be replaced by a generalized dot product that satisfies Mercer's theorem [3]. The only equation that changes is Eq. (8) which becomes

$$\mathbf{D}_{ij}=y_i y_j K(\mathbf{x}_i\mathbf{x}_j) \tag{13}$$

and the discrimination function

$$f(z)=\sum_{i=1}^{l}\alpha_i y_i K(\mathbf{z}\cdot\mathbf{x}_i)+b \tag{14}$$

We used a radial basis function neural network with kernel width σ, whose necessary number of kernels and their positions are found by the algorithm:

$$K(\mathbf{z},\mathbf{x}_i)=\exp\left(-\frac{|\mathbf{z}-\mathbf{x}_i|^2}{\sigma^2}\right) \tag{15}$$

5. SIMULATION STUDY

The channel used for our simulations was:

$$H_2(z)=-0.2052-0.5131\cdot z^{-1}+0.7183z^{-2}+0.3695z^{-3}+0.2052z^{-4} \tag{16}$$

The co-channel transfer function considered was:

$$H_3(z)=\lambda\left(0.6+0.8z^{-1}\right)\qquad\lambda>0 \tag{17}$$

Firstly, we provided a constant SIR=24 dB, and the noise power was changed to produce different SINR ratios . Then, the noise power was fixed to give constant SNR=24 dB, and the interfering signal power was changed by selecting choosing different values for λ [5]. The performance of our structure were measured in terms of $\log_{10}\mathrm{BER}$ (BER= Bit Error Rate), for 10^5 to 10^7 samples depending on SINR. We show in Fig. 3 the relationship between the bit error rate and the Wiener filter order (*d*=2) for different SINR conditions in two cases: a) SIR=24 dB; b) SNR=24 dB. This example suggest that a very small advantage could be gained by using an LTE which has an order greater than 5. We encountered in this case the problem of noise enhancement [2]. In the following simulations the Wiener filter order was chosen to be 5.

Next, we compare the performance of the Wiener filter, the RBF equalizer and SV equalizer under a variety of SINR conditions. For the nonlinear structures we chose *m*=4 and a delay $d=2$.The training sequence contained 640 samples (we need all at once for the SV method). The desired signal states are 256, and the RBF network was chosen to have 256 centers. For the SV structure we used $C=10^4$ and $\sigma^2=0.5$. The performance curves of Wiener filter, the RBF equalizer and SV equalizer are plotted in Fig. 4 (for fixed SIR=24 dB) and Fig. 5 (for fixed SNR =24 dB). Both equalizers seem to address the interfering signal differently from noise.

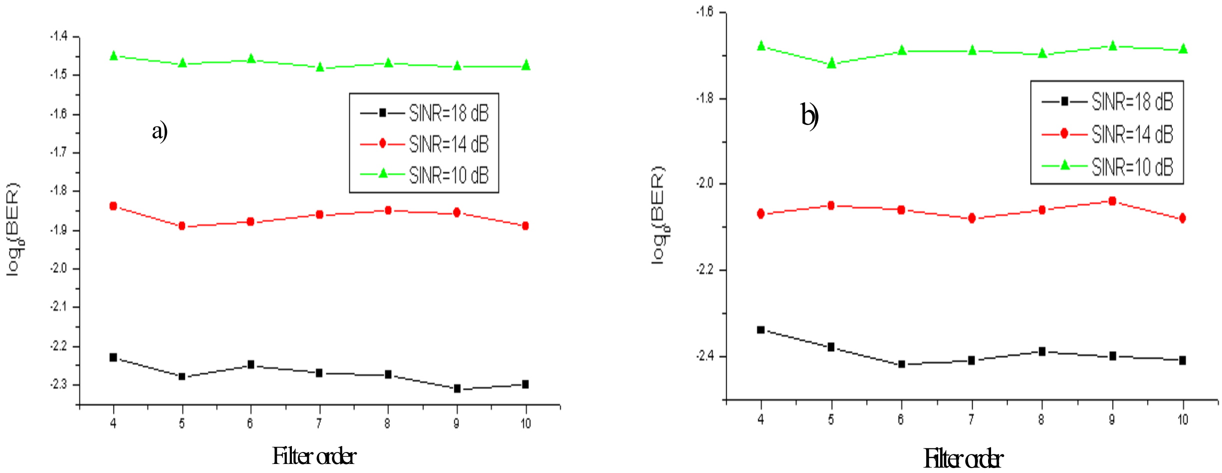
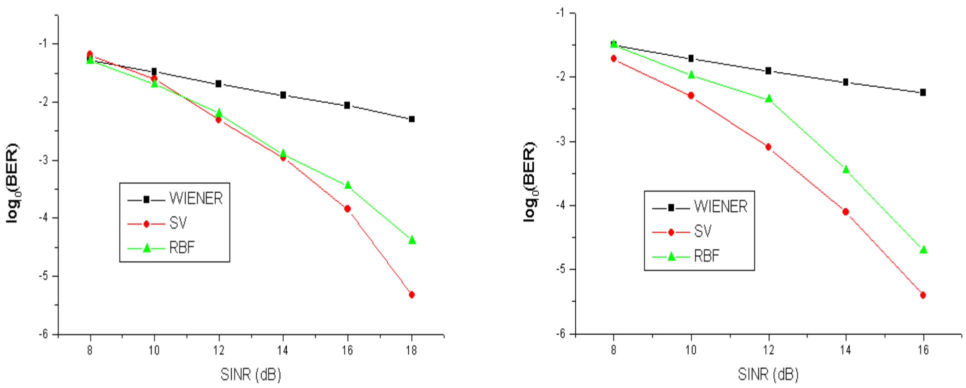


Fig. 3 Bit Error Rate versus Wiener filter order, d=2, a) SIR=24 dB b) SNR=24 dB

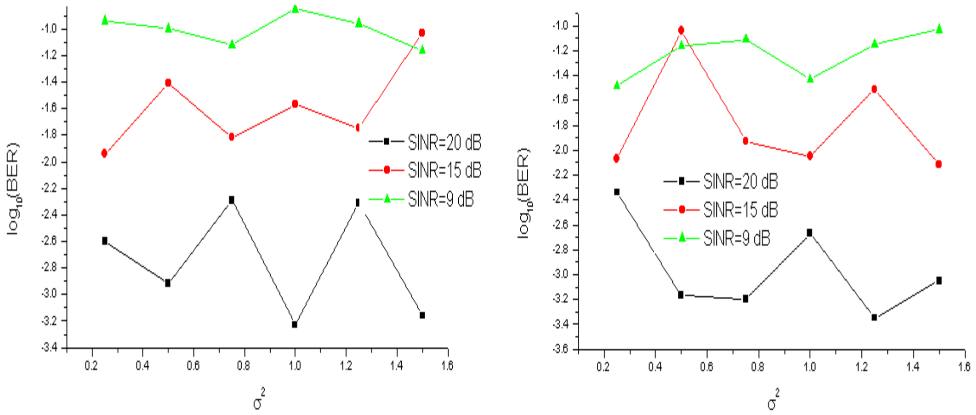


a) b)

Fig. 4 Comparison of performance. Channel H_2 with the interfering channel $H_3, d = 2$, a) SIR=24 dB, b) SNR=24 dB

The non-linear structures show a significant improvement over Wiener filter, especially for high SINRs. Clearly, the performance of RBF or SV equalizer obtained by changing SIR with a fixed SNR=24 dB are much better than by changing SNR with a fixed SIR=24 dB. We can infer the better performance of the SV method over the RBF technique using the classical k-means clustering method, particularly for high SINR ratio (see Fig. 4). The k-means algorithm is an unsupervised learning method based only on input training samples. The traditional k-means clustering algorithm can only achieve a local optimal solution, which depends on the initial locations of cluster centers. A consequence of this local optimality is that some initial centers can become stuck in regions of the input domain with few or no input patterns, and never move to where they are needed [7]. The SV method provides a better way of choosing the number and locations of RBF centers, suitable for our classification problem. It should be noted that computing the optimal Bayesian solution would be very costly (an RBF network with 8192 centers), and wasn't investigated in our study. The SV method fails to work satisfactorily for low SINRs because the errors were punished too much (Figs. 4). Better results could be obtained by lowering C and using other values for σ^2, depending on SINR conditions. Our simulations revealed also the significant sensitivity of this method with respect to strong levels of SINR.

The effect of σ^2 was investigated using the same system but with a different value of m (m=2). Figure 5 confirms that the equalization performance of the SV structure is sensitive to the parameter σ^2. Further studies must establish a proper value of this parameter depending on SINR conditions. A cross-validation technique could be used, although other heuristic methods would be indicated.



a) b)

Fig. 5 BER performance versus σ^2, *m*=2, *d*=2, a) SIR=24 dB, b) SNR=24 dB

CONCLUSIONS

The traditional view of RBF equalizers has been one where the clustering heuristic used for training them was considered very important. In contrast, the SV algorithm provides a way of choosing the number and locations of RBF centers, with the centers being those examples that are critical for the classification task. A comparative study of BER performance among theree techniques for channel equalization in hostile environments has been given. Our simulations showed the superiority in overcoming co-channel interference of SV machines over a RBF network using a classical k-means procedure, for high SINRs, and small training sequences. Our future work will focus on speeding training phase and on extending the current study to more general channel models.

REFERENCES

[1] G.D.Forney, ”Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference”, *IEEE Trans. Inform. Theory,* Vol.IT-18, No. 3, 1972, pp. 363-378

[2] S. Chen, B. Mulgrew, P. M. Grant, “A clustering technique for digital communications channel equalization using Radial Basis Function networks”, IEEE Transactions on Neural Networks, Vol 4, July 1993.

[3] Vapnik, V. “The Nature of Statistical Learning Theory”, Springer-Verlag, New York, 1995

[4] D. Tax, D. Ridder and R.P.W.Duin, “Support vectors classifiers: a first look”, Proceedings of the Third Annual Conference of the Advanced School for Computing and Imaging, ASCI, DELFT, June 1997

[5] S. Chen, B. Mulgrew, “Overcoming co-channel interference using an adaptive radial basis function equaliser”, EURASIP Signal Processing, vol. 28, pp. 91-107, 1992

[6] F. Albu, A. Mateescu, J. C. M. Mota, B. Dorizzi, “Adaptive channel equalization using neural networks”, *ITS'98*, August 1998, Sao Paulo, Brazil

[7] R. O. Duda, P. E. Hart, "Pattern Classification and Scene Analysis", John Wiley & Sons, 1973

[8] P. E. Gill, W. Murray, M. H. Wright, "Practical Optimization", Academic Press, 1981