Support vector classifiers: Application to digital communications channel equalization

1. Felix Albu, 2. Dominique Martinez
1. Faculty of Electronics & Telecommunications, 1-3 Bd. Iuliu Maniu, Bucharest
felix@pns.comm.pub.ro, Tel: 401 410 54 00
2. CNRS-LAAS, 7 av. Col. Roche, 31077 Toulouse, France
dmartine@laas.fr, Tel: 33 561 33 64 43

ABSTRACT

The Support Vector (SV) method is a novel type of learning machine, based on statistical learning theory. In support vector classifiers, the input vectors are mapped to a high dimensional space and are then separated by the optimal linear hyperplane[1]. In the RBF case, the SV algorithm automatically determines centers, weights and threshold that minimize an upper bound on the expected test error. The classical approach in building a radial basis function classifier consists of choosing the centers by some clustering techniques. The weights are usually found by the pseudoinverse method or by LMS algorithm[2]. We applied these techniques to channel equalization, viewed as a classification problem, for reconstruction of binary signals. It is well-known that the radial basis function network has an identical structure to the optimal Bayesian symbol-decision equalizer solution[3]. But the computational requirement to implement can be very high, as the full RBF Bayesian solution usually requires a large number of centers. We compared the results of equalization task obtained with classical RBF training with those obtained by the SV algorithm . We showed the superiority of SV techniques, especially for high SNR ratio.

1. INTRODUCTION

To addresss a classification problem, one often would like to use as many features as possible to improve the classification result. However, most classifiers suffer from what is called the small size effect or peaking effect. There is a certain optimal number of features beyond which performance will only degrade. This is because, as the dimensionality of the feature space grows, estimating densities of the data becomes increasingly harder for a finite number of samples[4]. Vapnik proposed a method for finding a hyperplane optimally dividing two classes, which does not depend on a probability estimation[1]. This optimal hyperplane is a linear decision boundary which separates the two classes and leaves the largest margin between the vectors of the two classes. It was observed that the optimal hyperplane is determined by only a small fraction of the data points, the so-called *support vectors.*
To find a non-linear classifier, it is possible to transform the vectors in *input space* to some high-dimensional *feature space.* This allows for more difficult problems to be solved and increases the feature space dimensionality, but hardly influences the classifier complexity. Furthermore, it was shown that the probability of making an error depends only on the number of these support vectors (and therefore on the complexity of the classifier) and the number of training vectors. In the next sections we present the Support Vector machines (we refer mostly to the paper by Cortez and Vapnik [5]), the RBF equalizers and some computer simulations showing the performances of these techniques on digital communication channel equalization.

2. SUPPORT VECTOR MACHINES

2.1 THE OPTIMAL HYPERPLANE

The optimal hyperplane is defined as the plane separating two separable classes in such a way that its margin is as wide as possible. 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

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

if the optimal hyperplane is

$$\mathbf{w}\cdot\mathbf{x}+b=0 \tag{2}$$

in which w can be written as

$$\mathbf{w}=\sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i,\quad \alpha_i\geq 0 \tag{3}$$

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

$$f(z) = \sum_{i=1}^{l} \alpha_i y_i (\mathbf{z} \cdot \mathbf{x}_i) + b \qquad (4)$$

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

$$W(\Lambda) = \Lambda^T \mathbf{I} - \frac{1}{2} \Lambda^\mathbf{T} \mathbf{D} \Lambda \qquad (5)$$

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

$$\Lambda \geq 0$$
$$\mathbf{Y}^\mathbf{T} \Lambda = 0 \qquad (6)$$

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

$$\mathbf{D}_{ij} = y_i y_j (\mathbf{x}_i \mathbf{x}_j) \qquad (7)$$

Maximizing Eq. (5) is a quadratic programming problem with linear constraints.

2.2 SOFT MARGIN HYPERPLANE

The concept of the optimal hyperplane is only suitable for the separable case[4]. If classes overlap Eq. (1) is modified to allow for errors:

$$y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 - \xi_i, \quad i = 1, ..., l$$
$$\xi_i \geq 0, \quad i = 1, ..., l \qquad (8)$$

and the following function can be minimized:

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i \qquad (9)$$

The quadratic programming problem now becomes a dual one:

$$W(\Lambda, \delta) = \Lambda^T 1 - \frac{1}{2} \left[\Lambda^\mathbf{T} \mathbf{D} \Lambda + \frac{\delta^2}{C} \right] \qquad (10)$$

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

$$0 \leq \Lambda \leq \delta 1$$
$$\mathbf{Y}^\mathbf{T} \Lambda = 0 \qquad (11)$$

The dot product can be replaced by a generalized dot product that satisfies Mercer's theorem [2]. The only equation that changes is Eq. (7) which becomes

$$\mathbf{D}_{ij} = y_i y_j K(\mathbf{x}_i \mathbf{x}_j) \qquad (12)$$

and the discrimination function

$$f(\mathbf{z}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{z} \cdot \mathbf{x}_i) + b \qquad (13)$$

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

$$K(\mathbf{z}, \mathbf{x}_i) = \exp\left(-\frac{|\mathbf{z} - \mathbf{x}_i|^2}{\sigma_2} \right) \qquad (14)$$

3. RBF EQUALIZERS

Channel equalization is a technique employed to combat the effects of intersymbol interference and noise which corrupt the transmission of signals across a communication channel (Fig. 1). The equalization aims to reconstruct the transmitted sequence with the minimum error probability, i. e. : $\tilde{s}(t) = s(t-d)$, where d is the delay.

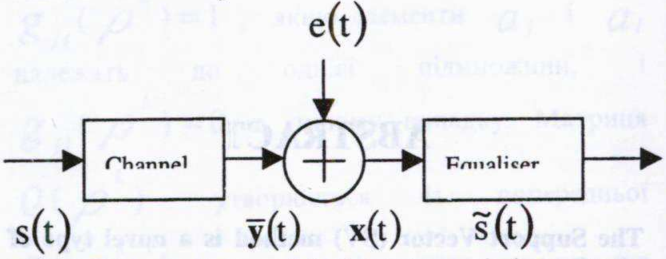

Fig.1 Discrete-time model of data transmission system

The channel is usually modeled by a FIR filter with the following transfer function :

$$H(z) = \sum_{i=0}^{n_h - 1} h_i z^{-i} \qquad (15)$$

where h_i is the channel impulse response components and n_h its length. In our study the transmitted symbol $s(t)$ is taken from the data set {±1}; it forms an i.i.d. sequence, and $e(t)$ is an additive white Gaussian noise with zero mean and variance σ_e^2 [3].

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 Euclidian distance between a center vector $\mathbf{c_i}$ and an input vector $\mathbf{x} = [x(t)\, x(t-1) ... x(t-m+1)]^T$. 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:

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i \Phi_i, \qquad (16)$$

where w_i are the weights.

It has been shown [3] that RBFN achieves an implementation of the optimal Bayesian equalizer if the channel is known and the parameters of the network are well chosen (i. e., the number of hidden neurons n is equal to the number of desired channel states: $n = 2^{m+n_h-1}$, the RBFN centers are placed at desired channel state vectors:

$\mathbf{c}_i = \left[\overline{y}_i(t)\ \overline{y}_i(t-1)\ldots \overline{y}_i(t-m+1)\right]^t$; the weights are chosen in the data set: {±1}, and $\sigma_i^2 = 2\sigma_e^2\gamma_i^2 = 2\sigma_e^2$, $i \in [1;n]$). Note that the RBFN structure can be complex when m and n_h are large.

In our work, the training of RBFN was done using a two-steps approach: in the first step a classical k-means clustering procedure was used to find the location of the centers and the widths were set at $\sigma_i = \dfrac{d_m}{\sqrt{2 \cdot n}}$, where $i \in [1;n]$ and d_m is the maximum distance between the chosen centers[6]. The k-means algorithm is an unsupervised learning method based only on input training samples. It partitions the input data set into n cluster centers. In the second step, the weights were trained using the least mean squares (LMS) algorithm.

4. SIMULATION STUDY

For the purpose of graphical display, the equalizer order is chosen equal to 2. Let the channel transfer function be $H_1(z) = 0.5 + z^{-1}$. The delay was 1.

Figure 2 shows 50 samples which form the data clusters and the SV centers(indicated by extra circles).

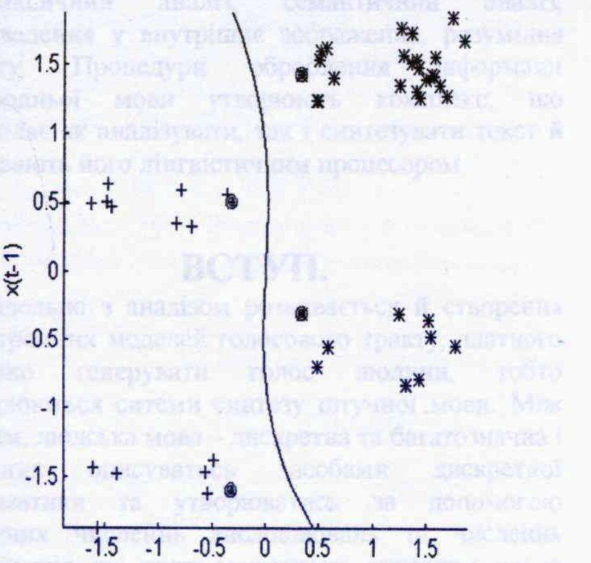

Fig.2 RBF centers automatically found by the SV algorithm (indicated by extra circles), and the decision boundary, $H_1(z) = 0.5 + z^{-1}$, SNR=17 dB, $\sigma^2 = 2$

The other channel used for various SNR's was:

$$H_2(z) = -0.2052 - 0.5131 \cdot z^{-1} + 0.7183 z^{-2} + 0.3695 z^{-3} + 0.2052 z^{-4} \quad (17)$$

We used a 2-64-1 RBF network, and the delay was chosen $\tau = 2$ [7]. The number of samples for training was 250 (wee need all at once for the SV method). The performance of our structures (RBF or SV equalizers) were measured in terms of $\log_{10}$ BER (BER= Bit Error Rate), for 200000 samples. We can see the superiority of SV method over a RBF network using this k-means clustering method especially for high SNR ratio (Fig.3). The SV method doesn't work well for low SNR's because the errors were too much punished. We used $C = 10^8$ and $\sigma^2 = 0.5$. Better results could be obtained by lowering C and using higher values for σ^2, directly related to SNR ratio. A cross-validation technique could be used although other heuristic methods would be indicated. It should be noted that with the clustering method presented in [3] the RBF structure could achieve the optimal Bayesian performance for a symbol by symbol equalizer. The convergence of this procedures requires for our particular case more than 600 samples. We also need a correct estimate of the channel order. The SV method provides a rigorous way of choosing the number and locations of RBF centers specifically selected for our classification problem.

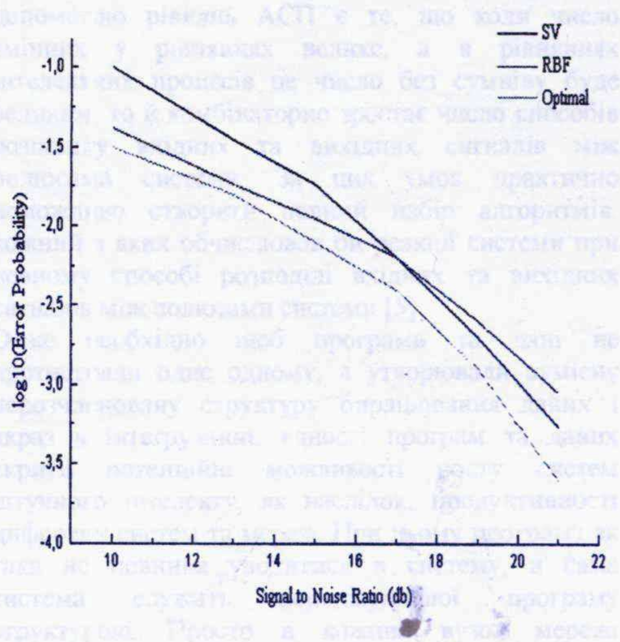

Fig.3 Comparison of performance. Channel H2, delay=2, m=2

5. CONCLUSION

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 the locations of RBF centers, where the centers are those examples that are critical for the classification task. The classical 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, failing to move where they are needed [8]. Our simulation showed the superiority of SV method over a RBF equalizer using a k-means clustering procedures for high SNR's ratio. Whether a RBF network can realize optimal equalizer solution depends crucially on positioning the centers at the desired channel states. This requires a great number of samples for training using a supervised or unsupervised k-means procedures and a correct estimate of the channel order. With a small number of training samples and no knowledge about the channel order, the SV method could become a suitable choice for digital communications channel equalization. In the future, our work will focus on attempting to enhance the speed in training and test phase while looking closely at their application to nonlinear channels.

REFERENCES

[1] Vapnik, V. "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995

[2] B.Mulgrew : "Applying Radial Basis Functions", IEEE Signal Processing Magazine, March 1996.

[3] 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.

[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] Cortes, C. and Vapnik, V. "Support Vector Networks", Machine Learning 20:273-297, 1995

[6] B. Dorizzi, J. C. M. Mota, F. Albu, " A step towards equalisation for radiomobile channel: Neural Networks and Variable Selection", Workshop on " Emerging Techniques for Communication Terminals", Toulouse, France, July 1997

[7] S. Chen ,B. Mulgrew, S.McLaughlin, "Adaptive Bayesian Equalizer with Decision Feedback", IEEE Trans. Signal Processing, 41 (9), pp. 2918-2927, 1993

[8] R. O. Duda and P.E. Hart, *"Pattern Classification and Sceen Analysis"*, Wiley, New York, 1973