A Novel Channel Equalizer Using Large Margin Algebraic Perceptron Network

Priti R. Hathy
Womens Polytechnic/CA, Bhubaneswar, India
Email: pritiranjanhathy@yahoo.com

Sasmita K. Padhy
ITER (SOA)/IT, Bhubaneswar, India
Email: chavisiba@rediffmail.com

Siba P. Panigrahi
KEC/EE, Bhubaneswar, India
Email: siba_panigrahy15@rediffmail.com

Prashant K. Patra
CET/CSE, Bhubaneswar, India
Email: hodcomputer@yahoo.co.in

***Abstract*- This paper proposes a novel control scheme for channel equalization for wireless communication system. The proposed scheme considers channel equalization as a classification problem. For efficient solution of the problem, this paper makes use of a neural network working on Algebraic Perceptron (AP) algorithm as a classifier. Also, this paper introduces a method of performance improvement by increasing margin of AP equalizers. Novelty of the proposed scheme is evidenced by its simulation results.**

***Index Terms*—Neural Networks, Algebraic Perceptron, Signal Recovery, Channel equalization**

I. Introduction

There is an ever-growing demand for high quality and high-speed wireless communication. One of the major limiting factors is inter symbol interference (ISI). Adaptive equalizers are used to reduce channel disturbances such as noise, ISI, CCI and adjacent channel interference (ACI), nonlinear distortions, fading, time-varying characteristics of channels, etc.

Researchers in [1-4] have developed some advances on equalizers. However, Neural network-based equalizers [5-8] have been proposed as alternative approaches to classical equalizers and provide significant performance improvement in a variety of communication channels. In spite of its good performances, NN models of channel equalizers have raised many controversial issues like: high value of complexities and lower margin. Though equalizers based on Support vector machines (SVM) [9, 10] increases margin, they aim only at maximizing the margin. This motivates this paper to go for an equalizer with large margin and to keep the network size down as well.

To avoid the above-mentioned difficulties in existing equalizers, this paper proposes an Algebraic Perceptron Neural network (APNN) with large margin. In recent literatures, researchers are dealing with equalization problem as an optimization problem and optimize them using soft and evolutionary algorithms [11-14]. Here in this paper, we solve equalization as a classification problem. The objectives of this paper can be outlined as: (1) To formulate problem of channel equalization as classification problem, (2) Placing Algebraic Perceptron (AP) network as a classifier, (3) Increasing margin of AP classifier. (Resulting equalizer termed here as APLM), (4) Comparing the results with existing large margin classifiers, i.e. with Support vector machines (SVM) and (5) The purpose of this paper is not to make APLM a substitute for the SVM, whose solution optimizes, but to consider its network size as well.

Advantages of this paper are that the proposed equalizer (APLM) tries to maximize margin with respect to the critical vectors rather than the whole data set.

Rest part of this paper is organized as: Section II formulates channel equalization as a classification problem. Section III gives a brief description on the AP basics. In section IV, we discuss Algebraic Perceptron Neural Network (APNN) as a classifier. Section V projects the improvements made using APNN. Section VI describes the simulation results and the paper is concluded in section VII.

II. Channel Equalization

The digital communication system considered in this paper is illustrated in the Fig. 1. The linear time invariant channel has been modeled as an FIR filter and its transfer function is given by [20]:

$$H(z) = \sum_{i=0}^{N} h_i z^{-i} \tag{1}$$

Where, h_i, is the channel parameter and N is order of channel. The output of the channel is corrupted with Additive White Gaussian Noise (AWGN), η_n.

The transmitted digital signal, $x(n)$, is an independent, equi-probable binary sequence. The task of the equalizer is to reconstruct the input symbol, $x(n-d)$, with the information contained in the received signal at the receiver, $y(n), y(n-1), \cdots, y(n-m+1)$, where m and d are the order and delay of the equalizer.

Input data, $x(n)$, is in form of binary sequence. Due to noise introduced in the channel, at the receiver a +1 may appears as -1 and vice versa. Hence, it is required to have a clear boundary between these two classes. Channel equalizer does this job of classifying the data. Hence, the channel equalization problem can be treated as classification problem and can avoid the need of channel inversion.

Manuscript received January 31, 2010; revised April 18, 2010; accepted May 11, 2010

doi:10.4304/jcm.5.8.637-645

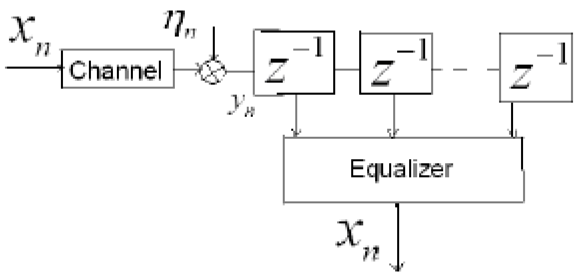

Figure 1: Model of Digital Transmission System

The channel input vector for mth order equalizer is given as:

$$X(n) = [x(n), x(n-1), \cdots, x(n-m+1)] \quad (2)$$

This can take $k = 2^m$ different values giving rise to k possible values of channel output vector as:

$$Y(n) = [y(n), y(n-1), \cdots, y(n-m+1)] \quad (3)$$

That can be divided into two classes as:

$$Y^+ = \{y(n) | x(n-d) = 1\} \quad (4)$$

$$Y^- = \{y(n) | x(n-d) = -1\} \quad (5)$$

III. ALGEBRAIC PERCEPTRON NEURAL NETWORK (APNN)

The AP is a binary pattern classifier. It projects the data onto a high dimensional feature space and constructs an arbitrary separating hyperplane between the positively labeled and the negatively labeled training data. As with other neural networks classifiers, the AP can be used as controller in electric vehicles. When compared with other traditional neural networks classifiers, such as the Multi-layer Perceptron, the Radial Basis Function Network or the Modified Probabilistic Neural Network, the AP can be trained much more quickly and easily.

According to Cover's theorem on the separability of patterns "ref" [15]:

> *An input space made up of nonlinearly separable patterns maybe transformed into a new feature space where the patterns are linearly separable with high probability, provided that two conditions are satisfied. First, the transformation is nonlinear. Second, the dimensionality of the feature space is high enough.*

This theorem has been the foundation for the AP. For the AP, two mathematical operations are involved. The first operation is the nonlinear mapping of multi-dimensional input vectors onto a high dimensional hypersphere in the feature space where linear separation is possible. This operation is hidden from both the input and the output.

The second operation is to find a separation on the hyper-sphere using a geometrical separation technique. The separation is achieved by an arbitrary hyperplane constructed using a sparse data representation.

Consider the problem of classifying data $\{x_i, y_i\}_{i=1}^N$, where x_i is the i^{th} m-dimensional input pattern vectors and $y_i \in \{\pm 1\}$ is the corresponding desired response or the target output.

A. Data Preprocessing

Because the AP operates in a hyper-sphere, some data preprocessing is required to project the data on to the hyper-sphere. The general equation of a decision surface in the feature space that does the separation is defined by:

$$f(w,b) = x_i \cdot w + b \quad (6)$$

Where, w is weight vector and b is bias.

The equation separates the positive and the negative examples:

$$x_i \cdot w + b \geq 1 \text{ for } y_i = 1 \quad (7)$$

$$x_i \cdot w + b < 1 \text{ for } y_i = -1 \quad (8)$$

However, because the AP performs separation in a hyper-sphere with the center of the hyper-sphere as the origin, the bias term, b, somehow has to be incorporated into the operation. This can be done by adding an additional dimension to the input vectors, that is, $x \in R^n$ is lifted to R^{n+1}. The input vector becomes $x' = (x_1, \cdots x_n, \lambda)$. So as to incorporate the bias, b, into the weight vector $wx = (w_1, \cdots w_n, b/\lambda)$. Here, λ is a scalar constant.

B. Normalization

The vectors are first mapped to a unit sphere. This procedure helps speed up the special geometrical operations on which the AP algorithm is based. First, the length of each vector in x' is to be the same. Normalizing the vectors can do this:

$$\bar{x}' = \frac{1}{\|x'\|} x' \quad (9)$$

This maps x' to a unit-sphere, $SR^{n+1} \subset R^{n+1}$.

C. Nonlinear Transformation

The vectors on the unit-sphere are taken to the feature space, V, through the mapping $\{\phi : S \to V\}$. An inner product kernel, k, is used to translate two vectors in the lower dimensional space, E, into inner products in the high dimensional feature space, V, as defined by:

$$k(x,y) = (\phi(x) \cdot \phi(y))$$
$$\Rightarrow k(x,y) = (\langle x,y \rangle_E)^P = \langle \phi(x), \phi(y) \rangle_V \quad (10)$$

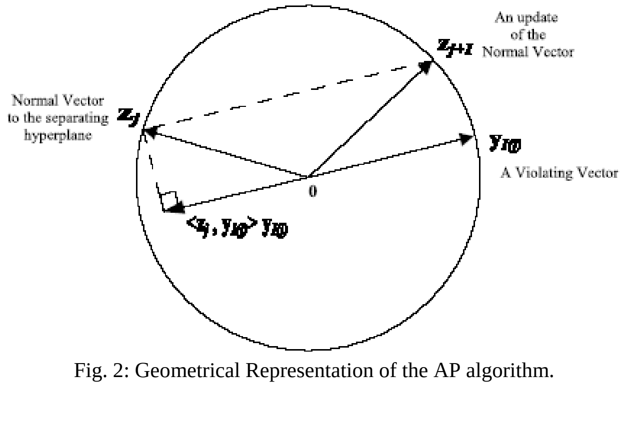

Fig. 2: Geometrical Representation of the AP algorithm.

$\phi(\cdot)$ is the transformation of a vector onto the feature space, V . However, $\phi(\cdot)$ is not calculated directly. The choice of kernel must be one defined in accordance with Mercer's theorem [16].

D. Training the Network

For the case of separable patterns, when the normalized vectors, $\phi(\bar{x}_i^{'})$ for $i = 1,\cdots N$, are multiplied by their respective desired output, $d_i \in \pm 1$, there exists a hemisphere that contains all the vectors. Training the network involves finding a hyperplane, through the origin of the sphere, which separates the hemisphere. A norm vector characterizes this hyperplane.

$\bar{x}_i^{'}$: The extended training input data, $i = 1,\cdots N$

d_i : The class label of $\bar{x}_i^{'}$

z_j : The normal vector of hyperplane at j[th] iteration in the feature space

y_i : The feature vector, where $y_i = \phi(\bar{x}_i^{'})d_i$

$I(j)$: Indicator function, which denotes the maximum violating element, $y_{I(j)}$, at iteration j; $I(j) \in \{1,\cdots,N\}$.

A violating vector, y_i , is a vector whose angle with z_j is greater than $\pi/2$, or equivalently an inner product $\langle z_j, y_i \rangle_V < 0$.

Starting from an arbitrary initial vector, z_0 , the position of z_j is evaluated iteratively by the AP algorithm. When separation has been achieved, that is, when the algorithm does not return a violating vector, the separation plane, which z_j is normal to, defines the boundary of the hemisphere.

The AP algorithm is formulated as follows:

1. Iteration $j = 0$, select any arbitrary initial normal vector z_j .
2. While $\exists i$ such that $\langle z_j, y_i \rangle$ <0, do
3. $y_{I(j)} = \arg ument\{\min\{\langle z_j, y_i \rangle, overall\, y_i, i = 1,\cdots N\}\}$
4. $z_{j+1} = z_j - 2\langle z_j, y_{I(j)} \rangle y_{I(j)},\ j = j+1$

Where $\langle z_j, y_{I(j)} \rangle y_{I(j)}$ is the projection of z_j onto the maximum violating vector $y_{I(j)}$, as shown in Fig 2.

The update z_{j+1} is also a vector on the hypersphere.

The above steps can be expressed recursively as follows:

$$z_j = z_0 - 2\sum_{k=1}^{j} \langle z_{k-1}, y_{I(k)} \rangle_V y_{I(k)} \quad (11)$$

To avoid evaluating vectors in the feature space, the Eq.(11) can be restated in terms of inner products via kernel functions.

$$\langle z_j, y_i \rangle_V = \langle z_0, y_i \rangle_V - 2\sum_{k=1}^{j} \langle z_{k-1}, y_{I(k)} \rangle_V \langle y_{I(k)}, y_i \rangle \quad (12)$$

or,

$$\langle z_j, y_i \rangle_V = \langle z_0, y_i \rangle_V - \sum_{q \in SV} \alpha_{q,I(k)} \langle y_{I(k)}, y_i \rangle \quad (13)$$

Where,

$$\alpha_{q,I(k)} = \sum_{k=1}^{j} \alpha_{q,k} \delta_{q,I(k)} \quad (14)$$

$$\alpha_{q,k} = -2\langle z_{k-1}, y_{I(k)} \rangle \quad (15)$$

$\delta_{q,I(k)}$ is the Kronecker Delta function.

The inner product in the feature space can be expressed as:

$$\langle y_{I(k)}, y_i \rangle_V = d_{I(k)} d_i k(x_{I(k)}, x_i) \quad (16)$$

The final $\langle z_j, y_i \rangle_V$, from Eq.(12) is expressed as a linear combination of the most violating vectors gathered at all iterations as shown in Eq.(13). These most violating vectors are the critical vectors for constructing the separating hyperplane. The general structure of the AP can be seen from this equation. The vector normal to the separating hyperplane is constructed by the sum of weighted critical vectors. The values $\alpha_{q,I(k)}$ are the coefficients for these critical vectors. Non-critical vectors have zero coefficients. The AP has an intrinsic property of finding vectors that lie closest to the decision boundary as critical vectors.

E. Soft Margin

There is no mechanism in the AP that takes care of the cases of non-separable patterns. In these cases, the algorithm continues to try to deal with the misclassified examples. In using the Gaussian Kernel, the dimensionality of the feature space becomes infinitely large. It is always possible to classify finite training data. As a general principle from learning theory, for two classifiers having the same training error, the classifier with smaller kernel parameter is more likely to perform better on unseen data [17]. Over fitting the data may give higher generalization error. In the setting of the problem (17), non-separable patterns, or the overlapping of data, are likely to be the results of noise. Separating the non-separable means that the classifier even tries to make "sense" of the effect of noise. For this reason, measures have to be taken in the AP to handle non-separability. Two approaches are available. In the first approach, the frequencies of violating samples are tracked. Frequent violating samples are eventually removed from training data [18]. In the formulation of the AP algorithm, the coefficients of the violating vectors are adjusted at each iteration. Non-separable samples are ones that frequently violate. This approach is based on observation and it does not properly penalize the measure of the violation on separability. However, it works well on the specific application like that of (1), when noise is high [18]. In the second approach, the coefficients of the critical vectors are bounded by a constant, C. This is done in the same way that the Lagrange multiplier in the SVM is bounded by the regularization parameter, C. The coefficients of non-separable samples grow in size at each iteration. Once the coefficient of a critical vector goes over the bound, C, the sample is removed from the training data. A similar method has also been proposed for dealing with the requirement of a soft margin classifier in [19].

F. Classification

Classification is performed according to Eq.(17). The decision criterion is the sign of the inner product between the test vector and z_j in the feature space. Positive signs indicate that the test vectors are labeled +1, and the inverse for the other classification. The structure of the AP is shown in Figure 3.

$$\hat{d}_i = sign\langle z_j, \phi(x_i)\rangle_V \quad (17)$$

IV. ALGEBRAIC PERCEPTRON CLASSIFIER (APC)

The AP has a definite advantage for problems where data density is high and ambiguity is low [19]. However, this is not the nature of the equalization problem where data clusters are distinct and noise is ever present. As a demonstration, the AP is used to solve the problem (17) discussed in section 2**.** The difference here is that the signals are binary taking values from $\{\pm 1\}$. Consecutive m-number of received symbols is the input to the AP for training and classification. Selecting the input dimension to be two, i.e. m = 2, the nature of the problem can be visualized more easily. Figure 4 shows the data distribution and decision boundaries constructed by AP in separate test runs. For the same set of data distribution, the decision boundaries are different in each test run. This demonstrates that the performance of AP can be unstable. The performance depends on where the decision boundary is located relative to the data distribution.

One of the ways to stabilize the performance of AP, according to the law of large numbers, is to make the size of the training sample, N, infinitely large. By observing the data distribution, it is intuitive that a better decision boundary should be constructed right in the middle of the gap between the data clusters of different classifications. In other words, in order to yield better results, the gap between the decision boundary and the data clusters from different classifications should be the same. The same argument should also apply to the case when the input dimension is more than two.

V. PERFORMANCE IMPROVEMENT BY INCREASING THE MARGIN

Mathematically, the reasoning behind the above mentioned intuition could be found in Vapnik's work on Statistical Learning Theory [20]. The principle of

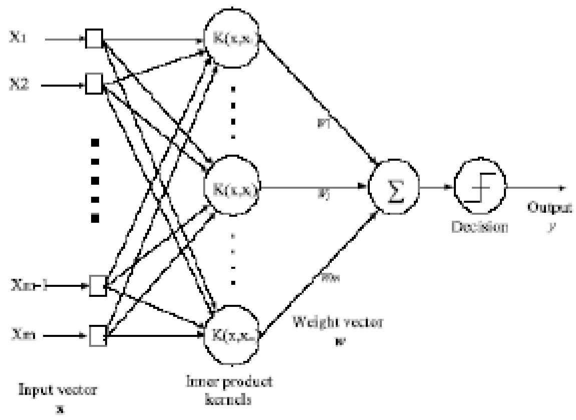

Figure 3: Architecture of the Algebraic Perceptron

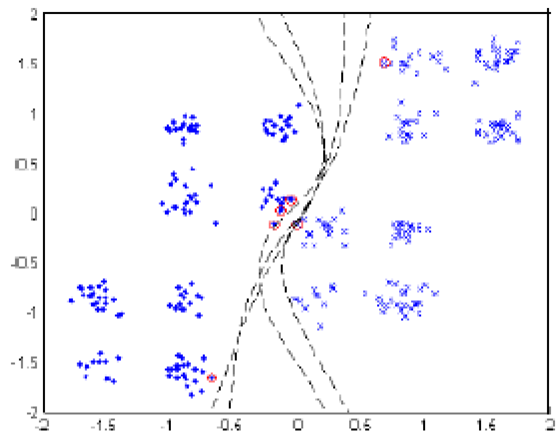

Figure 4: Different decision boundaries constructed by the Algebraic Perceptron during each run

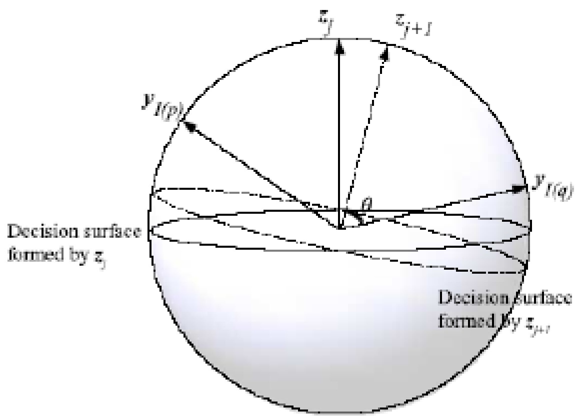

Figure 5: An update towards a better solution for the Algebraic Perceptron

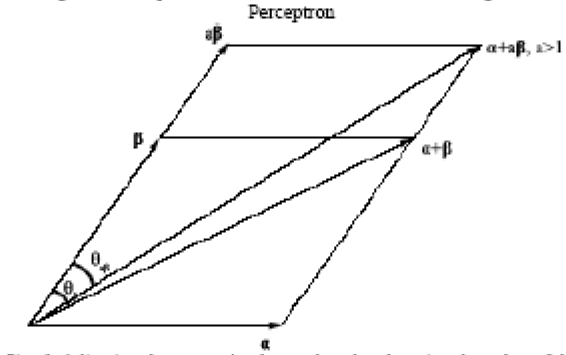

Fig. 6: Adjusting the separating hyperplane by changing the value of the coefficients of the critical vectors

Structural Risk Minimization reveals that minimizing the training error is only half way to minimizing generalization error. Generalization error is defined as the error rate of the learning machine after it was trained, when it is tested with examples from the same statistical distribution but not seen before. Minimizing generalization error is especially important to the application of channel equalization, because the training data can only be a small sample of the whole representation. Through better generalization it is possible to acquire reasonable results from a reduced set of training data.

The problem to be solved is this: given separation, find a better solution that gives a bigger margin. The method proposed is a simple adaptive method to increase the separation margin of the AP. This is an extension to the AP algorithm.

A. Measuring and Maximizing the Margin

The Euclidian distance can be used to determine the similarity between two vectors. Let x_i denote a m -by-1 vector of real elements.

$$x_i = \left[x_{i1}, x_{i2}, \cdots x_{im}\right]^T \quad (18)$$

The Euclidean distance between a pair of m -by-1 vector x_i and x_j is defined by:

$$d\left(x_i, x_j\right) = \left\| x_i - x_j \right\| = \left[\sum_{k=1}^{m}\left(x_{ik} - x_{jk}\right)^2\right]^{1/2} \quad (19)$$

Here, x_{ik} and x_{jk} are the k^th element of the input vector x_i and x_j respectively.

The similarity between the two vectors is defined as the reciprocal of the Euclidean distance $d\left(x_i, x_j\right)$. That is, the closer the two vectors are to each other, the smaller the Euclidean distance $d\left(x_i, x_j\right)$ will be. Another measure of similarity is the dot product or the inner product. Given two vectors of the same dimension, x_i and x_j, the inner product is defined by

$$\left\langle x_i, x_j \right\rangle = \left\|x_i\right\|\left\|x_j\right\|\cos\theta = x_i^T x_j = \sum_{k=1}^{m} x_{ik} x_{jk} \quad (20)$$

Here, θ is the angle between the two vectors. The relationship between Euclidean distance and inner product is shown in Figure 5. It can be seen that the computation of inner products is equivalent to carrying out geometrical constructions in terms of angles, length and distances. Now for the operation of AP, the vectors are normalized to have unit length, that is,

$$\left\|x_i\right\| = \left\|x_j\right\| = 1 \quad (21)$$

Eq.(19) can be used to write

$$d^2\left(x_i, x_j\right) = \left(x_i - x_j\right)^T\left(x_i - x_j\right) = 2 - 2x_i^T x_j \quad (22)$$

Eq.(22) shows that minimizing the Euclidean distance corresponds to maximizing the inner product. In other words, the inner product can be used as a measure for margin.

The margin is the key factor in the generalization analysis for such classifiers.

The margin of f on a labeled example (x, y) is the quantity $yf(x)$, that is, the amount by which the function f is to the correct side of the threshold. It can be thought of as an indication of the confidence with which f classifies the point x.

Figure 5 depicts that the separating hyperplane formed by z_j is a possible solution for the AP. For clarity, not all training data are shown in the figure. Only violating vectors from previous iterations, $y_{I(p)}$ and $y_{I(q)}$, are shown. The classifier can correctly classify the given training samples. However, the vector $y_{I(q)}$ is "closer" to the separating hyper plane than $y_{I(p)}$. If an update, $z_j + 1$, can be found that increases the margin, the generalization ability of the classifier will be improved, as is discussed in section IV. The margin here is defined as the "closeness" between the critical vectors and the separating hyper plane.

Now the separating hyper plane is represented by the normal vector z_j and the angles between z_j and all the training vectors $y_i, i = 1,2,\cdots,N$ are less than $\pi/2$. Since the cosine function between 0 and $\pi/2$ is a monotonic decreasing function and knowing that the vectors are of unit length, the "closeness" can be indicated by the cosine of the angle (or the inner product) between the normal vector and the critical vectors.

The further away a vector, y_i, is from the normal vector, z_j, the smaller the value of the inner product.

Maximizing the margin is equivalent to maximizing the angles between the normal vector and the critical vectors, provided that it still classifies correctly. Geometrically, the position of the normal vector $z_j + 1$ is best at the center of, or having the same Euclidean distance to, all the critical vectors.

B. Adjusting the Decision Boundary

The steps are to shift the position of the separating plane so that the margin is increased and yet the separation condition still satisfies. Because the normal vector is a linear combination of the critical vectors, the position of z_j can therefore be adjusted away from or towards the direction of a particular critical vector by changing the coefficients of that critical vector, $\alpha_{q,I(k)}$ in Eq.(13). This is demonstrated in Figure 6.

The idea is to locate and increase the value of the coefficient of the critical vectors of each class that are too "close" to the separating hyperplane. When the normal vector is equal in "distance" to all the critical vectors, the margin with respect to the critical vectors is maximized. Please note that the margin here does not mean the same thing as the margin in the SVM, where it is with respect to the entire data.

After the initial adjustment, the normal vector, z_j, is no longer of unit length. The inner product between the normal vector and the critical vector involves the angle and the length of the normal vector. However, this does not affect the approach of using the inner product to locate the critical vector whose coefficient is to be updated.

C. Adding New Critical Vectors

It may occur that during the process when the margin is being maximized, other vectors from training samples, y_i, could lie closer to the decision boundary than any of the critical vectors of the same label. In such a case, this particular vector should be included as a critical vector. Updating the coefficient of the vector using the same update method does this inclusion. The optimization of the margin should continue.

The following simple algorithm was devised:

- Step #1

Calculate the normalized variance of the inner products.

Stop if

$$\mathrm{var}\left(\left\langle z_j, y_{I(k)}\right\rangle_v\right) / norm\left(\left\langle z_j, y_{I(k)}\right\rangle_v\right) < \eta, \ for\ k = 1, \cdots N$$

or

$$m\arg in_{new} > m\arg in_{old}$$

else go to Step 2

- Step #2

Find the critical vector, $y_{I(l)}$, the inner product of which to the normal vector is smallest, i.e., $\min\left(\left\langle z_j, y_{I(k)}\right\rangle\right)$

Do

$$\left\{\alpha_{q,I(l)} = \alpha_{q,I(l)} + \varepsilon\right\}$$

- Step #3

Go to Step 1

Here,

$$m\arg in: \left\{\min\left(\left\langle z_j, y^{+}_{I(k)}\right\rangle\right) + \min\left(\left\langle z_j, y^{-}_{I(k)}\right\rangle\right)\right\}$$

N : Number of critical vectors

$I(l)$: Index of the critical vector whose distance to the decision boundary needs to be increased

ε : Step size can be fixed or variable. A good choice for variable ε may be; $\left\{mean\left(\left\langle z_j, y_{I(k)}\right\rangle\right) - \left\langle z_j, y_{I(k)}\right\rangle\right\}$ variable epsilon allows the solution to be found more quickly

η : A user defined error tolerance parameter

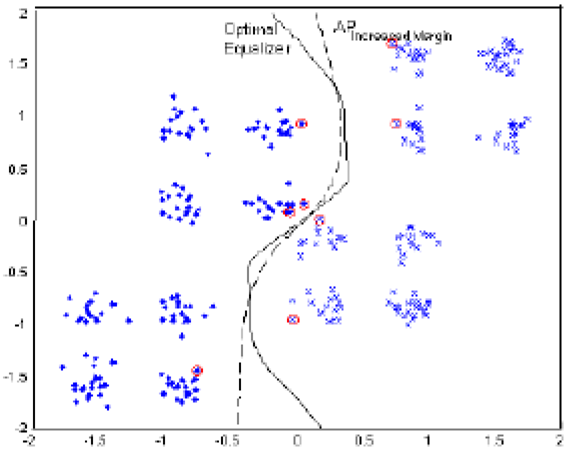

Figure 7: Decision boundaries of the APLM and the Optimal Bayesian equalizer

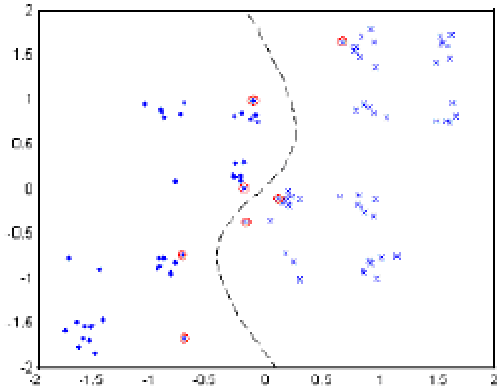

Fig. 8: Separation by APLM having only 100 training samples. SNR is 20dB.

D. The Improvement

The improvement to the generalization ability as a result of increasing the margin can be visualized by selecting the input dimension to be 2. This is shown in Figure 7. The figure is to be compared to previous results acquired using AP before the improvement, as shown in Figure 4. The decision boundary from the increased margin AP is compared with one constructed by an Optimal Bayesian Equalizer. The optimal equalizer is constructed given some prior knowledge about noise. It provides a good benchmark for the test equalizers.

It can be shown from Figure 7 that the decision boundary given by the increased margin AP follows quite closely the decision boundary provided by the optimal equalizer for this particular data distribution. Without actually performing Monte Carlo analysis to determine the error rate, the prediction is that the two will perform quite similarly.

One of the advantages of a large margin classifier is that it can be trained with fewer data to give the same result. To demonstrate this point using the APLM, Figure 8 shows the separation by the APLM with a mere 100 training samples. The decision boundary is similar to what is given by Figure 7.

VI. Simulation results

The methods discussed in previous sections are put to the test by running simulations. The channel used in the simulation is a discrete time linear phase channel modeled as FIR filter and the transfer function is given by:

$$H(z) = 0.35 + 0.87z^{-1} + 0.35z^{-2} \qquad (23)$$

The input data for the training and testing are chosen from the $\{\pm 1\}$ with equal probability. They are independent and identically distributed. The additive distortion introduced by the channel is modeled as white Gaussian noise with zero mean. For the training of the AP equalizer a binary sequence of 300 is randomly generated and fed to the digital channel. At the output of the channel white Gaussian noise with zero mean is added to introduce further distortion. This distorted sequence forms the input pattern for the AP equalizer. The proposed equalizer is trained to this pattern. The input dimension is chosen to be 4. The delay is set at 1. Gaussian kernels with small sigma values, $\sigma = 0.1$, were used. The purpose of this paper is to show the improvement in performance of having the margin increased in the AP, while other variables, such as the kernel type, are kept constant.

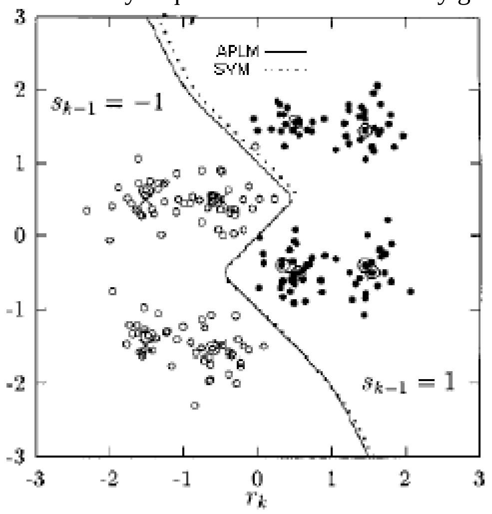

Figure 9: Decision boundary for SVM and APLM

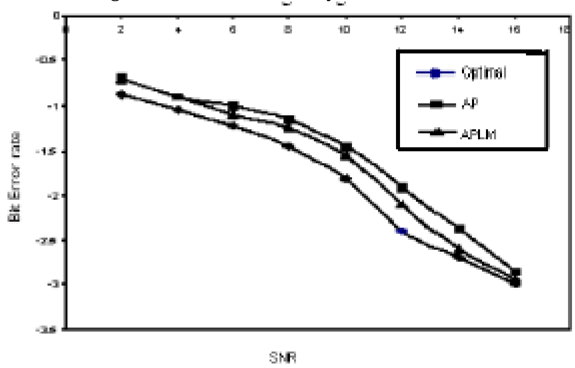

Figure 10: BER for optimal, AP and APLM equalizer

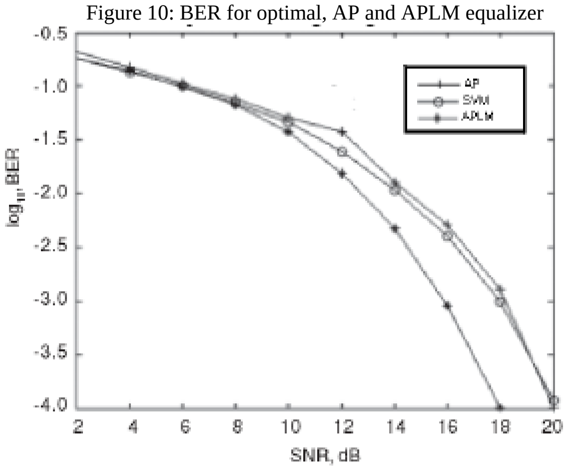

Figure 11: Log(BER) for optimal, AP and APLM

After the training phase is complete, the AP equalizer is tested with a new sequence of data generated from $\{\pm 1\}$ with equal probability. This data sequence is, then, passed through the channel to introduce the ISI. White Gaussian noise is added to the output of the channel to introduce the additive distortion in the signal. This distorted binary sequence is applied to the input of the proposed AP equalizer.

B. Comparing with the Support Vector Machine

There is a fundamental difference between this approach and the approaches by other large margin classifiers, such as the Support Vector Machine (SVM). The difference is that the SVM aims only at maximizing the margin, whereas the APLM tries to keep the network size down as well. In the SVM, the margin is maximized with respect to the complete training data. The network size is dependent on the parameter settings and the structure of the training data. Unless the data structure is known before hand or that the network is retrained with different parameter settings, it is often difficult to know whether the network size can be further reduced to give a more efficient run time implementation. The APLM on the other hand increases the margin with respect to only the critical vectors. In the cases, which are encountered in channel equalization, the decision boundary formed by increasing margin of the critical vectors approximates the decision boundary produced by the SVM. However in the former, the number of critical vectors can be monitored and be kept low. This is especially useful if the run time efficiency is an important consideration. However, the reduction in network size comes with a slight performance penalty. The argument for persisting with it even with the loss of performance is as follows. In practice, the equalizer is trained with a noisy data set. To be really precise, in determining the maximal margin boundary the effect of noise is generalized. Therefore, it is reasoned that by relaxing the precision by a small extent it should generate similar results.

C. Results and analysis

In order to study the performance of proposed equalizers AP and APLM, we have compared the results with SVM and optimal Bayesian equalizer. This performance study was based on four parameters: (1) Decision boundary, (2) Error performance in terms of Bit Error Rate (BER) and $\log_{10}(BER)$, (3) Network size and (4) stability issues.

Decision boundaries were studied in figures 4, 7, 8 and 9. Error probability configured in figures 10 and 11.

Similarly, Network size is evidenced from table 1. Stability is demonstrated through an example at the end of this section.
Study of figures 9 reveals that, there is no large difference between decision boundaries framed by SVM and proposed APLM. Also from figure 7, though it seems a difference in decision boundary between Bayesian optimal equalizer and proposed APLM, a keen study shows there is no difference in pattern classification of output signal.
Figure 10 and 11 shows that for SNR of less than 12 dB, the APLM does not possess any advantage. This is because at this high noise level, data clusters are loosely bounded. The "gaps" between the data clusters become narrow. The room for increasing the margin becomes restricted. However, as the noise level decreases, the improvement in performance allows the APLM to get close to the performance of the Optimal Equalizer and SVM. While, on the other hand, the AP struggles to stay error free.
The APLM equalizer cannot achieve the same results that the Optimal Equalizer did at high noise level. This is because by observing the input data alone, the APLM could not distinguish between the effects due to noise and effects due to the channel response, whereas for the Optimal Equalizer the noise distribution is known a-priori. For the APLM, the size of the network is the number of critical vectors required to perform a particular classification. It translates directly to the calculation complexity to classify one input. Table 1 shows that size of the APLM when it is required to equalize the channel defined by Eq.(23) when the SNR is 20dB. As a reference, the table also lists the number of centers required by an optimal Bayesian equalizer.
The number of centers for the optimal equalizer is the number of possible states.
What Table 1 demonstrates is that the size of the APLM network does not grow exponentially with the increase of input dimension.
The optimal Bayesian equalizer, on the other hand, does. This has to do with the way the critical vectors are chosen in the APLM. One of the ways to reduce the size of the network is to choose centers that are close to the decision boundary. This is exactly what was done in choosing critical vectors for the APLM. In order to maintain the accuracy of the equalizer, when the input dimension is increased, the number of training samples also needs to be increased.
This is because the dimensionality of the input space rapidly grows and leads to a point where the training data is sparse and does not provide a good representation of the mapping.
This is to be noted that, a mere 300 training data were adequate to train the network. This is a distinct advantage for using the APLM as an equalizer.

D. Example: Nonlinearity

The problem of stability exists because the AP only constructs an arbitrary decision boundary between the binary patterns.

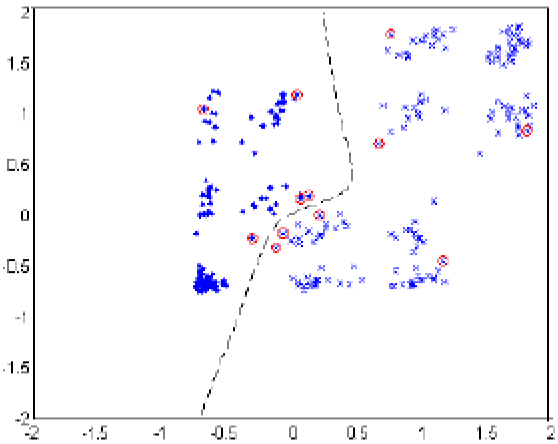

Fig. 12: Decision boundary for channel with Nonlinearity

TABLE I:

SIZE OF NETWORKS VS. DIFFERENT INPUT DIMENSIONS

Input dimension	2	3	4	5	6	10
AP_{LM}	7	18	30	50	60	70
Bayesian	16	32	64	128	256	4096
SVM	16	28	54	98	176	2664

For classifying problems with low data density, like the case for channel equalization, where the space between data clusters can be relatively large, the performance becomes unstable. The fix to the problem is to reduce the arbitrariness in constructing the decision boundary. According to the principle of structural risk minimization, it is better that the margin between the decision boundary and the two classifications is maximized. A method that increases the margin of the solution produced by the AP was proposed. The problem of stability in using the AP as a nonlinear channel equalizer has been addressed via this example. In simulations it was shown to give superior and more stable results.
Using APLM on channels with nonlinear distortions should not affect the improvement in performance over the AP. This is demonstrated in the following example.
Nonlinearity in the channel is simulated:

$$y(n) = x(n) + 0.2x^2(n) - 0.1x^3(n) + noise(n) \quad (24)$$

For visualization purposes, the effect of the nonlinearity on the data distribution is demonstrated in Figure 12, as compared to Figure 7.

VII. Conclusion

This paper proposed two novel equalizer structures, AP and APLM. Advantages of this paper can be outlined as: (1) The purpose of this paper is not to make APLM a substitute for the SVM, whose solution optimizes, but to consider its network size as well, (2) The APLM tries to maximize margin with respect to the critical vectors rather than the whole data set, (3) This is to be noted that, a mere 300 training data were adequate to train the network of proposed APLM, (4) Significant reduction in network size as compared to Optimal Bayesian equalizer and SVM equalizers, (5) Significant improvement in performance as compared to SVM based equalizers, (6) This paper also addressed stability concerns for large margin classifiers through an example and in simulations proposed APLM was shown to give superior and more stable results.

ACKNOWLEDGMENTS

The authors wish to thank the reviewers and the editor for constructive suggestions towards improvement in quality of this article.

REFERENCES

[1] M. Botros Shenouda and T N. Davidson, 'A Framework for designing MIMO systems with decision feedback equalization or Tomlinson-Harashima precoding', IEEE Journal on Selected Areas in Communications, Volume 26, Issue 2, Feb'2008, pp. 401 – 411.

[2] Manuel A. et.al, 'Sequential Monte Carlo methods for complexity-constrained MAP equalization of dispersive MIMO channels' Elsevier Signal Processing 88, 2008, pp. 1017–1034.

[3] S. Chen, et.al, 'Adaptive Bayesian space–time equalization for multiple receive-antenna assisted single-input multiple-output systems' Elsevier Digital Signal Process. (2007).

[4] Y. R. Zheng, et.al, Frequency-domain channel estimation and equalization for shallow-water acoustic communications. Physical Communication 3 (2010) 48-63.

[5] Chagra W, et al. 'Equalization with decision delay estimation using recurrent neural networks', Advances in Engineering Software 36, 2005, pp: 442–447.

[6] Ling Z.,et al. 'MIMO Channel Estimation and Equalization Using Three-Layer Neural Networks with Feedback', Tsinghua science and technology, Vol 12, no.6, Dec2007, pp 658-662.

[7] J. Lee & R. Sankar, 'Theoretical derivation of minimum mean square error of RBF based equalizer' Signal Processing 87, 2007, pp 1613–1625.

[8] H. Zhao & J. Zhang, 'Functional link neural network cascaded with Chebyshev orthogonal polynomial for nonlinear channel equalization', Signal Process. (2008).

[9] S. Chen, S. R. Gunn, and C. J. Harris, "The Relevance Vector Machine Technique for Channel Equalization Application" IEEE Transactions on neural networks, Vol 12, N. 6, Nov'2001.

[10] Daniel J. Sebald and James A. Bucklew, Support vector machine techniques for nonlinear equalization, IEEE Trans. Signal Processing 48 (2000), no. 11, 3217-3226.

[11] D.P. Acharya & G. Panda, Y.V.S. Lakshmi, "Effects of finite register length on fast ICA, bacterial foraging optimization based ICA and constrained genetic algorithm based ICA algorithm" *Digital Signal Processing, In Press, Corrected Proof, Available online 12 August 2009.*

[12] Babita Majhi & G. Panda, "Development of efficient identification scheme for nonlinear dynamic systems using swarm intelligence techniques" *Expert Systems with Applications, Volume 37, Issue 1, January 2010, Pages 556-566.*

[13] M. A. Guzmán, et.al, "A novel multiobjective optimization algorithm based on bacterial chemotaxis" *Engineering Applications of Artificial Intelligence, In Press, Corrected Proof, Available online 5 November 2009.*

[14] M.A. Shoorehdeli, et.al, "Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended kalman filter" *Fuzzy Sets and Systems, Volume 160, Issue 7, 1 April 2009, Pages 922-948.*

[15] B.E. Boser, I.M.Guyon, and V.N. Vapnik, A training algorithm for optimal margin classifiers, Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, D. Haussler, editor (1992), 144-152, Pittsburgh, PA, ACM press.

[16] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (1998), 121-167.

[17] C.J.C. Burges and B. Scholkopf, Advances in neural information processing systems 2, ch. Improving the Accuracy and Speed of Support Vector Learning Machines, pp. 375-381, MIT Press., Cambridge MA, 1997.

[18] T. Cacoullos, Estimation of a multivariate density, Ann. Inst. Statist. Math 18 (1966), no. 2, 179-189.

[19] I. Cha and S.A. Kassam, Channel equalization using adaptive complex radial basis function networks, IEEE, Selected Areas in Communications 13 (1995), no. 1, 122-131.

[20] V.N. Vapnik, Statistical learning theory, New York: Wiley, 1998.

Mr. Priti Ranjan Hathy, presently working with Women's Polytechnique, Bhubaneswar and working for his Ph.D.

Mrs. Sasmita Kumari Padhy, presently working with ITER, SOA University, Orissa, India.

Dr. Siba Prasada Panigrahi, presently working with Krupajal Engineering College, Bhubaneswar.

Prof. (Dr.) Prashant Kumar Patra is now working as Professor & Head, Department of CSE at College of Engineering & Technology, Bhubaneswar, Orissa, India.