BLIND EQUALIZATION OF NONLINEAR CHANNELS USING HIDDEN MARKOV MODELS

Kristina Georgoulakis and Sergios Theodoridis
University of Athens
Department of Informatics, TYPA Buildings, 15771 Athens Greece
e-mail: kristina@di.uoa.gr, stheodor@di.uoa.gr

ABSTRACT

A novel blind channel equalizer is proposed which is suitable both for linear and nonlinear channels. The proposed equalizer consists of the following three steps: a) identification of the clusters formed by the received data samples, via an unsupervised learning clustering technique, b) labeling of the identified clusters, by using a Hidden Markov Modeling (HMM) of the process and c) channel equalization by means of a Cluster Based Sequence Equalizer. The performance of the equalizer is investigated for a variety of channels (minimum/non-minimum phase, linear/nonlinear channels).

1 Introduction

Intersymbol Interference (ISI) is a major impairment in today's high bit rate communications systems. Channel equalizers, used in the receiver part, aim to suppress the effect of ISI. In most of the cases, the communications channel is unknown, and the design of the equalizer is performed on the basis of a known training sequence of information bits. However, there are many cases that transmission of a training sequence is not possible or desirable. This mode of equalizer design is known as *'blind'*.

Blind channel equalization is a challenging task and has been the focus of intense research effort. Recently, an interest has risen on approaches based on data clustering techniques, [1], [2]. A major advantage of such approaches is that no explicit channel modeling is required, that makes them attractive when nonlinear channels are involved, [4]. A cluster-based blind channel estimation algorithm consists of two steps : a) data clusters are first identified via an unsupervised learning technique and b) labeling of the identified clusters is achieved, by unraveling the information hidden in the sequence of received data, [1], [2]. When the channel estimation task is completed, a Cluster Based Sequence Equalizer, [3], can be employed to provide signal detection.

In this paper, a novel cluster - based blind channel estimation procedure is proposed. The novelty of the technique is in the way the identified clusters are labeled. Labeling is performed using a Hidden Markov Modeling (HMM) of the estimation process and by relating data clusters to HMM states. The probability of each data cluster to correspond to a specific label is treated as the unknown parameter of an HMM learning problem, which is estimated by the Baum-Welch (B-W) algorithm.

2 Channel Equalization as a classification task

System description

Let us consider the received signal $g(t)$ of an ISI and noise impaired system. In the general case, $g(t)$ can be written as

$$g(t) = f(I(t), I(t-1), ..., I(t-L)) + w(t) = c(t) + w(t) \quad (1)$$

where $f(\cdot)$ is the function representing the channel action, $I(t)$ is an equiprobable sequence of transmitted data, $w(t)$ is an AWGN sequence and $c(t)$ is the noiseless channel output sequence. In the special case of a linear channel, $g(t)$ can be written as

$$g(t) = \sum_{i=0}^{L} h(i)I(t-i) + w(t) \quad (2)$$

where $h(i)$ is the channel impulse response.

Clustering Based Sequence Equalization

The equalizer, placed at the receiver part, aims at recovering the transmitted sequence of information bits $I(t)$, based on the corrupted received sequence $g(t)$. In [3] a Clustering Based Sequence Equalizer (CBSE) is proposed which treats equalization as a classification task. This method focuses on the clusters, which the received data form. The received data samples are clustered around specific points whose number and constellation shape is determined by the spread of the channel and the impairments characteristics [4]. Assume, for example, D successive received data, in the absence of any distortion, corresponding to M^D possible points in the D-dimensional space (M is the size of the transmitted data alphabet). Each point corresponds to one of the M^D possible realizations of the sequence of transmitted bits: $(I(t), ..., I(t-D+1))$. If received data are corrupted by AWGN, then the randomness of noise leads to the formation of a *cluster* around each point. The existence of ISI causes a movement and an increase of the number of clusters. Specifically, for a channel with spread $L+1$ and for an M-ary alphabet the data vector $\mathbf{g}(t) = [g(t), g(t-1), ..., g(t-D+1)]^T$ leads to the formation of $C = M^{L+D}$ clusters. Each cluster is represented by a suitably chosen *representative*, which is the noiseless channel response vector in the D-dimensional space, i.e., $\mathbf{c}(t) = [c(t), ..., c(t-D+1)]^T$, with $\mathbf{c}(t) \in \{c_i, i = 1, ..., C\}$, each c_i corresponding to one of the possible values of the sequence: $(I(t), ..., I(t-L-D+1))$.

Due to the interdependence that ISI imposes on successive received data, only specific transitions among the different clusters are possible. Thus, CBSE employs a Viterbi type procedure dictated by the specific transitions among clusters. Assuming that data are treated in groups of D, then, a M^{L+D-1} state trellis can be considered, where the state $S(t)$ at time t labels the channel memory: $S(t) = (I(t-1)I(t-2)...I(t-L-D+1))$.

In the Viterbi trellis diagram, the transition from a state $S(t-1)$ to a state $S(t)$ corresponds to the emission of a specific cluster representative, indicated by the sequence of bits formed by the current state and the new information bit transmitted. We call this sequence of bits *label* and we denote it by:$X(t) = (I(t)I(t-1)...I(t-L-D+1))$.

For the completion of the Viterbi Algorithm procedure an appropriate distance metric is adopted (e.g., Euclidean or Mahalanobis) in order to measure the distance among the received data and the representatives of the clusters.

Training of the clusters is equivalent to finding the correspondence between clusters representatives and labels. In supervised equalization, during the training period, a sequence of known information bits is transmitted and each cluster representative is computed by a simple averaging of all the data vectors, $\mathbf{g}(t)$, belonging to the respective cluster. However, in blind mode, no training sequence is available and, thus, an appropriate unsupervised technique should be employed to provide the training of the clusters. In the proposed method, the unsupervised clusters training is performed in two steps a) clusters representatives identification and b) labeling of them.

3 Unsupervised clustering

The task of clusters identification can be performed using various unsupervised clustering techniques. Typical examples are : the isodata algorithm, [2], the neural gas network, [1], a recursive dynamic programming procedure, [5], etc. In this paper, the dynamic programming procedure described in [5] is adopted.

In [2] and in [1] the clustering procedure is performed using the two dimensional observation space $(D = 2)$ and the corresponding data vector is $\mathbf{g}(t) = [g(t)g(t-1)]^T$. The two-dimensional representation of clusters is required in [1] in order to determine the possible transitions among different clusters. In [2], the two-dimensional approach is needed for avoiding the problem of uncertainty arising from clusters which overlap in the one-dimensional space. In the proposed algorithm, *one dimensional exploitation of the clusters is adequate,* i.e. $(\mathbf{g}(t) = [g(t)])$, since a) clusters transitions information is not needed and b) overlapping clusters do not effect the performance of the proposed method.

4 HMM and clusters labeling

Once the clusters representatives have been identified, their corresponding labels have to be determined. For this purpose, a discrete observations HMM is constructed utilizing the already identified values of clusters representatives. *The discrete observations HMM formulated is characterized by the following elements* [6]:

1) The states of the model, which in our case are:

$$S(t) = (I(t-1)...I(t-L)) \tag{3}$$

where $I(t)$ is the i.i.d. sequence of transmitted symbols and L is the channel memory. For an M-ary alphabet, the number of the states is $N = M^L$, that is : $S(t) \in \{1, ..., N\}$.

2) The state transition probability distribution $A = a_{ij}$, where

$$a_{ij} = P[S(t+1) = j | S(t) = i], \quad 1 \le i, j \le N. \tag{4}$$

In our case, a_{ij} are known and are equal to $1/M$, for an allowable transition, or equal to zero, for a not allowable transition. For every allowable transition $(a_{ij} = 1/M)$ a specific noiseless channel output occurs. In other words, *each state transition specifies a cluster label.* The cluster labels are specified by:

$$X(t) = (I(t)...I(t-L)) \tag{5}$$

with, $X(t) \in \{1, ..., C\}$, $C = M^{L+1}$. Assume, for example, that $L = 1$ and $M = 2$, then, transition from state 1 $(I(t-1) = -1)$ to state 1 $(I(t) = -1)$, defines the label : $([I(t)I(t-1)] = [-1-1])$, which corresponds to a specific cluster.

3) The distinct observation symbols per transition $V = \{v_k\}$, $k = 1...C$. These are assumed to be equal to the clusters representatives.

4) The observation symbol probability distribution in states transition i to j. This parameter, in our case, corresponds to the probability of a specific cluster representative (symbol) to correspond to specific label (states transition). For simplicity, we use indices of labels and not of states transitions, since there is a unique correspondence between labels and states transitions, as stated earlier. Thus, this element is defined as :

$$b_n(v_k) = P[v_k \;\; observed \, | S(t) = i, S(t+1) = j] =$$
$$P[v_k \;\; observed \, | X(t) = n], \, 1 \le n, k \le C, \;\; 1 \le i, j \le N \tag{6}$$

where n corresponds to the label that uniquely specifies a state transition: from i to j.

5) The initial state distribution $\pi_i = P[S(1) = i]$ for $1 \leq i \leq N$

Due to the randomness of noise, the received data, $g(t)$, in the channel output have not discrete values (see eq. 2). In order to agree with the described discrete values HMM, the noisy received data samples are quantized to the value of the closest representative, and the sequence of quantized data is denoted by the symbol $y(t)$. Hence, *the quantized received data form the discrete observations of the HMM.* And, according to the discrete observations model, we define as $b_n(y(t)) = b_n(v_k) \times \delta(y(t), v_k)$, with $\delta(y(t), v_k) = 1$ if $y(t) = v_k$, and zero otherwise.

In the proposed blind channel estimation algorithm, clusters' labeling can be treated as an HMM learning problem; that is, to model the unknown probability of a specific cluster to correspond to a specific label as an unknown parameter of the HMM, and then seek for the optimal parameters of HMM which best match the given observations sequence. A usual practice to handle the learning problem in HMM, is the maximization of the probability of the observation sequence of length T: $Y = (y(1), ..., y(T))$ given the model parameters (θ), that is the probability $P(Y|\theta)$. The EM (Expectation - Maximization) algorithm (or else the B-W reestimation formulae) is a commonly used numerical scheme which estimates the unknown parameters of an HMM. The resulting ML estimate is given by : $\hat{\theta} = argmax_\theta P(Y|\theta)$. In our case, we define :

$$\theta = \{b_n(v_k), \quad n, k = 1, ..., C\} \tag{7}$$

These probabilities reveal the label of the clusters and are expected to converge to 1, if a specific symbol ($y(t) = c_k$) corresponds to a specific label (n) otherwise they converge to zero.

According to the above described model settings, the maximization of $P(Y|\theta)$ by means of the B-W algorithm, leads to the labeling procedure described in the followings:

Initialization

Take a block of T received data $g(t)$ and quantize them to their closest representative. Assign (randomly) the representantives values to the HMM parameters: v_k, for $k = 1, ..., C$. Set $N = M^L$, $C = M^{L+1}$, $b_n(v_k) = 1/C$ and $\pi_1 = i$.

Set $a_{ij} = P[S(t+1) = j|S(t) = i] = 1/M$ if there is an allowable transition from state i to j, otherwise, set $a_{ij} = 0$. Set $\alpha_1(i) = 1$ for the known initial state i and 0 otherwise, $\beta_{T+1}(j) = 1$ for $j = 1...N$.

Main part - Recursion

1) Use the forward recursion [6] to calculate $\alpha_t(j)$ for $t = 2, ..., T+1, j = 1, ..., N$

$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) P[S(t) = j|S(t-1) = i] b_n(y(t-1))$

2) Use the backward recursion [6] to calculate $\beta_t(i)$ for t=T,T-1,...,1, i=1,...,N

$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) P[S(t+1) = j|S(t) = i] b_n(y(t))$

3) Calculate the the probability of being in state i at time t, and state j at time $t+1$, given the model and the observation sequence [7] :

$\xi_t(ij) = \frac{\alpha_t(i) a_{ij} b_n(y(t)) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_n(y(t)) \beta_{t+1}(j)}$

4) Finally, use the reestimation formulae to update the:

$b_n(v_k) = \frac{\sum_{t=1,\, y(t)=v_k}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \xi_t(i,j)}$

Repeat the recursion for the next block of T (quantized) data, until convergence.

Note that, in the procedure described above, in order to simplify the presentation the assumption that the initial state, $S(1)$, is known, is made [6]. The generalization of the results to the case where $S(1)$ is unknown can be easily made : the algorithm is able to estimate prior probabilities $P[S(1) = i]$ [6],[7].

5 Simulations

Example 1 At first, a simple linear channel is considered to clarify the operation of the proposed algorithm. The assumed channel has transfer function $H(z) = 1 + 0.5z^{-1}$ and $L = 1$. The SNR is 20 dB. Transmitted data are assumed bipolar ($I(t) = \pm 1, M = 2$). The clustering- HMM algorithm is as follows : a) The unsupervised algorithm identifies the clusters representatives (i.e., $c_1 = 0.4926$, $c_2 = -0.49$, $c_3 = -1.503$, $c_4 = 1.498$, with the number of data used: $K = 200$). b) Next, the HMM is formed. We set N=2, $C = 4$ and the number of data processed per recursion: $T = 20$. The matrix of $b_n(v_k)$ converge, in about 8 recursions, to the following table:

Label		Center
	I(t) I(t-1)	$c_1 c_2 c_3 c_4$
1	(-1-1)	0 0 1 0
2	(-1 1)	0 1 0 0
3	(1-1)	1 0 0 0
4	(1 1)	0 0 0 1

From this table we can conclude the clusters' labeling, i.e., the representative c_1 has label [1 -1] etc. Once the channel estimation procedure is completed the CBSE can be used for signal detection.

Example 2 - Overlapping Clusters In this experiment, data are assumed binary ($M = 2$) and the channel is : $H(z) = 0.5 + 0.7z^{-1} + 0.5z^{-2}$. This is a high-ISI channel, resulting in overlapping clusters in the one dimensional space [6]. Thus, instead of the 8 clusters expected to appear ($C = 2^{D+L} = 8$, for $L = 2$ and $D = 1$), only 6 clusters are observed. In this case, our algorithm is as follows : a) Clusters representatives estimation results in the following representatives (SNR = 20 dB, $K = 200$), ($c_1 = .7$, $c_2 = .3$, $c_3 = -.7$, $c_4 = -.3$, $c_5 = 1.7$, $c_6 = -1.7$). b) The number of HMM states is N=4, and $C = 6$. The matrix of $b_n(v_k)$, for $n = 1, ..., 8$, $k = 1, ..., 6$ converge to the following matrix (in about 7 recursions, with $T = 40$):

Label		Center
	I(t)I(t-1)I(t-2)	$c_1 c_2 c_3 c_4 c_5 c_6$
1	(-1-1-1)	0 0 0 0 0 1
2	(-1-1 1)	0 0 1 0 0 0
3	(-1 1-1)	0 0 0 1 0 0
4	(-1 1 1)	1 0 0 0 0 0
5	(1-1-1)	0 0 1 0 0 0
6	(1-1 1)	0 1 0 0 0 0
7	(1 1-1)	1 0 0 0 0 0
8	(1 1 1)	0 0 0 0 1 0

From this matrix we recover the labeling of clusters, i.e., cluster representative c_1 (with value : .7) has two labels, the [-1 +1 +1] and the [+1 +1 -1], this is a result of the symmetry of the channel which actually causes the overlapping of clusters. The other labels of the representatives are determined in the same manner.

From this example, we see that *the situation of overlapping clusters does not cause problems in the suggested algorithm.* It should be emphasized that overlapping clusters (usually resulting from symmetric channels) can lead the cluster-based algorithms of [1] and [2] in incorrect clusters labeling. *The algorithms of [1] and [2] need the information of a starting point for their initialization.* This starting point is a cluster that jumps to itself and usually is the cluster which corresponds to a label of same symbols (i.e. $([+1+1+1])$ if $L = 2$). However, in the overlapping clusters case, a cluster (which corresponds actually to 2 or more overlapping clusters with different labels) can seem to jump to itself although its label is not as described before. Thus, a false starting point is assumed. Since, the subsequent procedure of clusters labeling is dependent on the correctness of this starting point - cluster, is apparent that this situation can cause serious problems in the algorithms of [1] and [2].

Example 3 - Nonlinear channel Consider for example the channel with $H(z) = 0.34+0.87z^{-1}+0.34z^{-2}$ and with the nonlinear function: $g(t) + 0.05g(t)^2 - 0.1g(t)^3$. In figure 1, we see the performances of three equalizers : a) the proposed cluster-HMM blind equalizer, b) a CBSE with exact mapping available and c) a classical MLSE with channel estimation achieved through an RLS algorithm. From the figure is apparent that the performance of the proposed equalizer is the same with the performance of the supervised CBSE. Moreover, the performance of the proposed equalizer is substantially better compared to the performance of the conventional MLSE using training sequence.

It should also be noted that *the proposed equalizer is able to treat every type of channels nonlinearities.* In contrast, the equalizer of [1] and [2], due to the special way of initialization that they adopt, constraint their use only in monotonic nonlinearities.

The performance of the proposed algorithm has been investigated on a variety of channels and for a number of signaling schemes. All the results indicate the

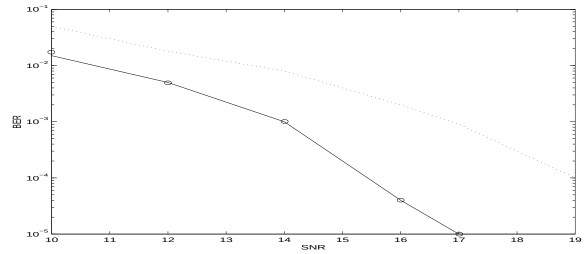

Figure 1: Performance achieved for nonlinear channel (example 3) ' o ' : proposed blind equalizer, ' - ': CBSE with known clusters centers, '.' : MLSE with linear channel estimator

robustness of the proposed scheme. The adoption of one dimensional model in the unsupervised clustering procedure gives complexity and convergence benefits to the proposed equalizer compared with the corresponding schemes of [2], [1]. In the above experiments channel order is assumed known. The problem of unknown channel order is under investigation.

References

[1] Y.J.Jeng, C.C.Yeh "Cluster Based Blind Nonlinear - Channel Estimation" IEEE Tr. On Signal Processing Vol.45, No 5, May 1997, pp.1161-1172

[2] S.Theodoridis, K.Georgoulakis "Efficient Clustering Techniques for Supervised and Blind Channel Equalization in a Hostile Environment" Proc. Eusipco 1996, Trieste, Italy, September 1996, pp.611-614

[3] K.Georgoulakis, S.Theodoridis "Efficient Clustering Techniques for channel equalization in hostile environments" Signal Processing vol.58, 1997, pp.153-164

[4] S.Theodoridis, C.F.N. Cowan, C.P.Callender, C.M.S.Lee "Schemes for equalization of communications channels with nonlinear impairments" IEE Pro. Commun. vol 142, 1995, pp.165-171

[5] N.Sheshadri "Joint Data and Channel Estimation Using Blind Trellis Search Techniques" IEEE Tr. on Communications Vol 42, No 2/3/4, 1994 pp.1000 - 1011

[6] G.K.Kaleh and R.Vallet "Joint parameter estimation and Symbol Detection for linear or nonlinear unknown channels" IEEE Tr. on Communications vol.42, No7, Jul 1994, pp.2406-2413

[7] L.R.Rabiner "A Tutorial on HMM and Selected Application in Speech Recognition" Proceedings of the IEEE, vol.77, No.2, Feb.1989.