ADAPTIVE CHANNEL EQUALIZATION USING CONTEXT TREES

O. E. Kelly *D. H. Johnson*

Department of Electrical and Computer Engineering
Rice University MS 366, Houston, TX 77005-1892, USA
oekelly@rice.edu *dhj@rice.edu*

ABSTRACT

The maximum likelihood sequence estimator is the optimal receiver for the inter-symbol interference (ISI) channel with additive white noise. A receiver is demonstrated that estimates sequence likelihood using a variable order Markov model constructed from a crudely quantized training sequence. Receiver performance is relatively unaffected by heavy-tailed noise that can undermine the performance of Gaussian based algorithms such as decision feedback equalization with gradient based (LMS) adaptation.

1. THE PROBLEM

We consider the problem of decoding binary symbols across a linear ISI channel contaminated with additive white noise. Given discrete-time observations of the channel output $r_1^n \stackrel{\text{def}}{=} r_1, \ldots, r_n$, we wish to estimate the transmitted bit sequence $B_1^n \stackrel{\text{def}}{=} B_1, \ldots, B_n$ where $B_k \in \{+1, -1\}$. The minimum probability of error solution is found by maximizing the posterior probability $\Pr[B_1^n | r_1^n]$ over possible bit sequences. Equivalently we may maximize the joint probability $\Pr[r_1^n, B_1^n] = \prod_{k=n}^{1} \Pr\left[r_k, B_k \left| r_1^{k-1}, B_1^{k-1}\right.\right]$. This optimal solution assumes knowledge of the underlying noise distribution and the channel's filtering characteristics. In many problems, neither of these are known precisely. When the ISI is linear and the noise is Gaussian, adaptive receivers can be designed to cope with channel uncertainties. In virtually all other situations, no universal approach to receiver design is known, and using the linear-ISI-Gaussian-noise receiver can yield poor performance levels.

We have described a technique built on type-based detection theory for receiving direct-sequence-coded bit streams when no ISI is present under general white noise conditions [1]. Here, we extend this type-based approach to the ISI channel, retaining the white noise assumption. Let γ be a quantizer that maps observations r_k to a finite alphabet $\mathcal{A}$ so that $\tilde{r}_k \stackrel{\text{def}}{=} \gamma[r_k] \in \mathcal{A}$. In the current work, we model the random process $(\tilde{r}_k, B_k)$ as a Markov process using the universal *context tree* model described in [2] to estimate the conditional probability law $\Pr\left[\tilde{r}_k, B_k \left| \tilde{r}_1^{k-1}, B_1^{k-1}\right.\right]$ from training data. The estimated probability law is used in conjunction with the Viterbi algorithm to find the probability maximizing bit sequence for the incoming stream of decision statistics. Figure 1 shows the basic function of the context tree equalizer after training. In [3, Section 6.6], Proakis notes that probabilistic solutions to the ISI problem can outperform the decision feedback equalizer (DFE) but have, so far, the disadvantage of being parametric and of also requiring a large number of computations per received signal. Being both non-parametric and computationally inexpensive, tree-based probability estimation techniques overcome these difficulties and offer access to the suggested performance advantages.

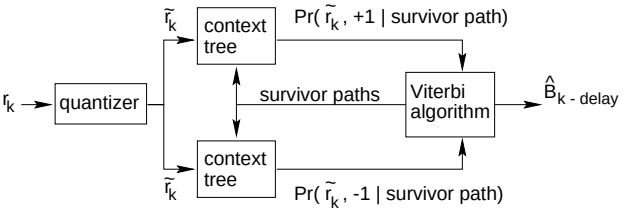

Figure 1: Context tree equalizer.

2. DISCRETE FORMULATION OF ISI CHANNEL

The transmitted signal for sending bits B_k at rate $1/T$ by antipodal signaling with waveform $g(t)$ is

$$s(t) \stackrel{\text{def}}{=} \sum_k B_k g(t - kT).$$

Let us assume a linear channel [impulse response $h(t)$] with additive white receiver noise $w(t)$ that may or may not be Gaussian. The message is received as $r(t) = w(t) + \int s(\tau)h(t-\tau)\,d\tau$. The channel response $h(t)$ is zero after time $t = L$. Decision statistics arise from sampling the output of a filter, having impulse response $m(t)$,

This work supported by NIMH. O.E. Kelly is now with WINLAB, Rutgers University.

at the symbol rate $1/T$.

$$r_i \stackrel{\text{def}}{=} \int r(t)m(t-iT)\,dt$$
$$= w_i + \sum_{k=i-\lceil L/T \rceil}^{i} B_k h_{i-k} \qquad (1)$$

where $h_{i-k} \stackrel{\text{def}}{=} \int\int g(\tau - kT)h(t-\tau)m(t-iT)\,d\tau\,dt$ and $w_i \stackrel{\text{def}}{=} \int w(t)m(t-iT)\,dt$. Equation (1) is the discrete formulation of the ISI channel.

The current observation r_i is a random variable correlated with previous observations but also dependent on bits which are *not* observable. Thus, $\{r_k\}$ alone is not a Markov process of any order. However by assuming a random description of the bitstream, we can form the sequence (r_k, B_k) that admits a Markov description [4]. In particular, assume the bitstream is q^{th} order Markov (typically $q = 0$). Samples r_i are sufficient statistics for the decision problem if $m(t) = (g * h)(-t)$ [see, e.g., [3]]. For that choice of $m(t)$, the noise samples w_i are correlated up to lag $\lceil L/T \rceil$ and the process (r_k, B_k) has Markov order $\max(2\lceil L/T \rceil, q)$. However, if the matching filter $m(t)$ is orthogonal to itself at time shifts of $T, 2T, \ldots$, then (r_k, B_k) has order $\max(\lceil L/T \rceil, q)$. Although r_i is real-valued, if the densities are smooth, it is reasonable to estimate a Markov model for the quantized process $(\tilde{r}_k, B_k)$ that takes its value in $\mathcal{A} \times \{+1, -1\}$. See [5] for a related approach to modeling the relationship between correlated finite alphabet sequences.

Equation (1) describes sampling the ISI channel with *one* sample per bit period. If p samples are taken per bit period, we denote the j^{th} observation of bit B_i by $r_{ip+j}, j = 0 \ldots p-1$. We may suppose that such observations arise from sampling using one of p different matched filters.

$$r_{ip+j} \stackrel{\text{def}}{=} \int r(t)m_j(t-iT)\,dt, \quad j = 0 \ldots (p-1)$$
$$= w_{ij} + \sum_k B_k h_{i-k,j}$$

The process $x_{ip+j} \stackrel{\text{def}}{=} (r_{ip+j}, B_i)$ is a cyclo-stationary Markov process. That is, for each fixed j, the distribution of x_n depends on a fixed number of preceding values.

In Section 4, we consider a transmitter response that is a length-p train of square pulses with polarities defined by a spreading sequence in $\{+1, -1\}^p$. In that case, the *chip*-matched filters are identical up to an integer number of chip-duration shifts $m_j(t) = m_0(t - jT/p), j = 0 \ldots (p-1)$.

3. CONTEXT TREE

A context tree is the underlying data structure in several methods [2, 6] for estimating the conditional probability mass function of a finite alphabet Markov process based on an observed *training* sequence. A context tree source, Figure 2a, is an alternate representation for a Markov source, Figure 2b. The probability mass function for a symbol, x_{n+1}, resides on the tree leaf specified by the recent history or *context* of the process: $x_n, x_{n-1}, \ldots$. The context tree source has fewer defining parameters because, where possible, contexts that provide no useful distinction are merged.

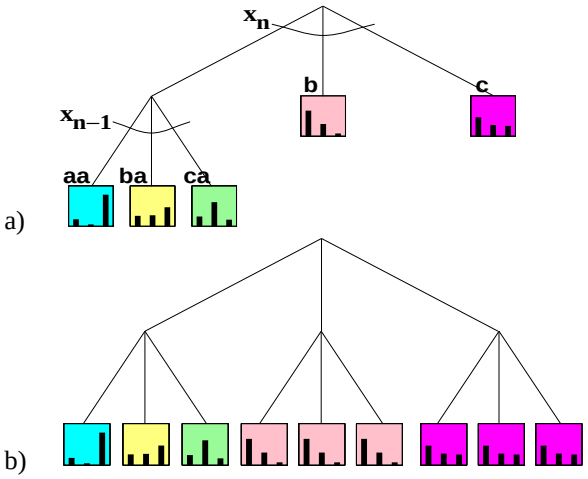

Figure 2: Context tree source. Figure a) shows the suffix tree representation of a context set $\mathcal{S} = \{aa, ba, ca, b, c\}$ defined over the alphabet $\mathcal{A} = \{a, b, c\}$. For the context tree a), the number of defining parameters is $|\mathcal{S}|(|\mathcal{A}| - 1) = 5 \times (3-1) = 10$ whereas the equivalent Markov source has 18 parameters. Histograms located at tree leaves are the conditional distributions of x_{n+1} where the conditioning sequence, $x_n\ x_{n-1}$, is represented by the path that leads to the leaf.

Estimating a context tree source, or training, consists of counting the number of occurrences in the training sequence of all possible subsequences of length less-than or equal to some maximum order D. Counts are arranged in a tree as in Figure 3, analogous to the trees in Figure 2. The second stage of training consists of examining the accumulated tree and, in the approach of Weinberger et al. [2] selecting a particular best tree by context merging, or, in the approach of Willems et al. [6] assigning probability by a convex combination of contributions from all possible subtree shapes of depth less-than or equal to D. The two approaches are called *pruning* and *weighting* respectively. The computation required to estimate the probability of a length n sequence using either method is $\mathcal{O}(nD)$ [7], or $\mathcal{O}(D)$ per received signal. We have shown the true Markov order is proportional to the duration of the channel response L and therefore the technique is on par with decision feedback equalization that requires a number of filter taps (hence computation) also proportional to L. For the cyclo-stationary

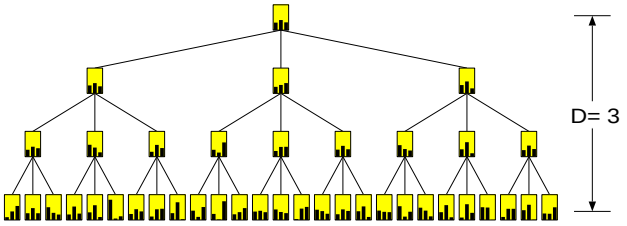

Figure 3: Context tree data structure. The context tree serves as a data structure into which we may accumulate counts of all transitions present in a sequence up to a chosen order D. The raw statistics are counts of the occurrence of letter a preceded by context s where s is a string of letters of length less-than or equal to D.

Markov process we estimate one context tree for each delay value $j = 0, \ldots, p-1$.

4. SIMULATION

Figure 4 shows the modulation scheme and ISI channel that is the focus of our study. We compare the performance of the context tree equalizer with an adaptive decision feedback equalizer (designed assuming Gaussian noise and linear ISI) in the presence of additive white noise with three different marginal distributions and over a range of SNR/bit values from -1 dB to +9 dB. The marginal distributions were Gaussian, Laplacian, and 99% Gaussian + 1% Cauchy. The spreading sequence is a 31 chip Gold code (0x04B3E375). The ISI channel filter is a first order recursive filter with time constant equal to 5 chips. When training occurs, each detector uses the same 500 bit training sequence. Twenty different training sequences were used at each SNR level. A different number of test bits were simulated at each SNR level to attain approximately the same coefficient of variation in the probability estimates (CV≈ 0.3): The number ranged between 380 test bits at SNR=-1 dB and 22,920 test bits at SNR=+9 dB.

The decision feedback equalizer consists of a 72-tap feedforward section to match and whiten the incoming chip sequence and a 3-tap feedback section to cancel the ISI of the last three bits. Filter coefficients are adjusted by the LMS algorithm to minimize residual errors of the filter in (soft) bit prediction $(\widehat{B}_i - B_i)^2$. Bit decisions are sign($\widehat{B}_i$).

For the context tree, observations r_{ip+j} are first mapped to a six letter alphabet $\tilde{r}_{ip+j} \in \mathcal{A} = \{0,1,2,3,4,5\}$ by a scalar quantizer. A new quantizer is calculated at each SNR level based on the empirical cumulative distribution of the data from 250 bits so as to place 1/6 of the data in each bin. For training, each quantized observation is paired with the current bit value $x_{ip+j} = (\tilde{r}_{ip+j}, B_i)$ and accumulated into the j^{th} context tree. A 16-state Viterbi algorithm uses estimated probability $\widehat{\Pr}\left[x_{(i-1)p+1}^{ip} \middle| x_{(i-4)p+1}^{(i-1)p}\right]$ as its path metric, evaluating that quantity once for each *survivor path* in its maximum likelihood search strategy. Note that in $x_{(i-1)p+1}^{ip}$ and $x_{(i-4)p+1}^{(i-1)p}$ the observations $\tilde{r}_{(i-1)p+1}^{ip}$ and $\tilde{r}_{(i-4)p+1}^{(i-1)p}$ remain fixed while the Viterbi algorithm specifies bit combinations B_i and B_{i-4}^{i-1} corresponding to different survivor paths. See Figure 1. In practice we use context tree pruning [2] to provide probability estimates to the Viterbi algorithm because the weighting method of [6] does not generalize well to large alphabets. Neither the context tree nor the DFE were operated with decision feedback training/adaptation.

Results of the simulation are shown in Figure 5. In Gaussian noise the CTE is competitive with the DFE, requiring approximately one more dB of SNR to achieve the same probability of error performance. However in heavy-tailed noise, the CTE appears to have equal or better performance.

Execution times for the CTE using pruning is approximately 130 times that of the DFE. Both detectors are implemented in a combination of interpreted and compiled Matlab code (m-files and mex-files). It is unclear whether the discrepancy would remain in an optimized implementation. The CTE is easily parallelized by providing a separate process for each of the p context trees that comprise the cyclo-stationary context tree. That approach would reduce the CTE execution time to $130/31 \approx 4.2$ times that of the DFE.

5. DISCUSSION

Although the choice of the scalar quantizer is somewhat arbitrary, we emphasize that the CTE is "model-free" and therefore able to adapt to interference situations beyond the scope of the underlying assumptions in the design of the LMS-trained DFE algorithm. The lack of robustness in the DFE algorithm, clearly demonstrated by the failure of the DFE to accommodate non-Gaussian noise environments, is evident by its performance sensitivity to noise distribution. Our algorithm's robustness does not pay a large computational cost: The calculation effort required by the context tree equalizer is of the same order as the DFE.

The current implementation uses context tree pruning and based on the training size (500) and alphabet size ($6 \times 2 = 12$), the algorithm [2] does not allow depth greater than 3 chips. In our simulations, the channel response is non-zero for about three time-constants or 15 chips. The DFE thus had a natural advantage in this simulation because its feedforward section was of sufficient length to accommodate the channel response. Ideally, the tree methods would be able to incorporate *a priori* knowledge of the channel in the same way that the number of filter taps in a DFE reflects

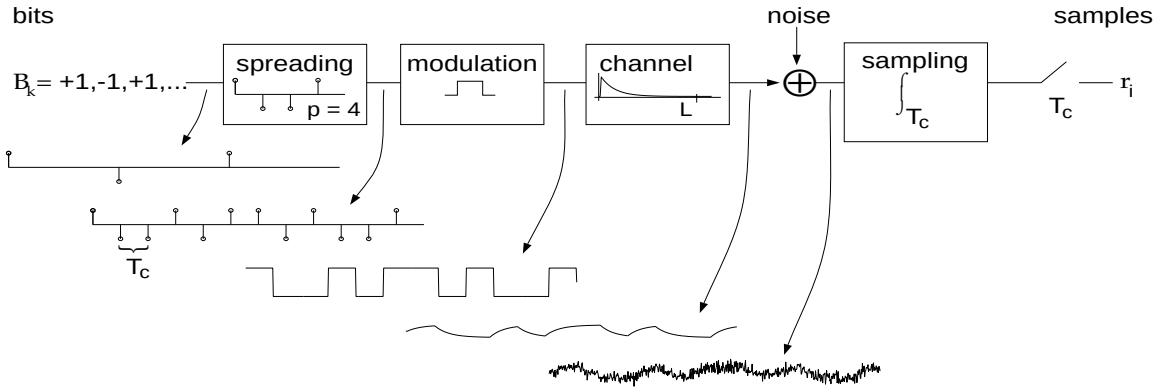

Figure 4: Intersymbol interference channel. We consider antipodal signaling with ± 1 valued signature sequences over a first-order linear ISI channel with additive noise. Length $p = 4$ spreading code is shown for clarity; simulations use $p = 31$. The sampling filter is a simple boxcar integrator over the chip period $T_c \stackrel{\text{def}}{=} T/p$.

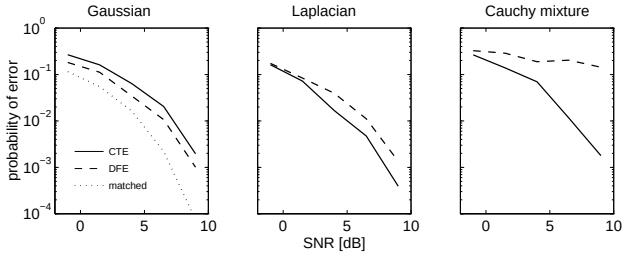

Figure 5: Simulation results. The first panel compares the performance of the context tree equalizer to an adaptive decision feedback equalizer and to the optimal detector in the presence of additive white Gaussian noise. At the same probability of error, the context tree equalizer requires approximately one dB more SNR/bit. However, in Laplacian noise (middle panel), performance is comparable. The third panel shows performance in the presence of additive white mixture noise that is 99% Gaussian and 1% Cauchy. The adaptive DFE is not robust to the heavy-tailed noise.

the designer's prior knowledge. That may be possible using the weighting methods described in [7], but only if they are combined with CTW algorithms that have low redundancy for large alphabets (results in [6] hold only for $|\mathcal{A}| = 2$).

6. REFERENCES

[1] D. H. Johnson, Y. K. Lee, O. E. Kelly, and J. L. Pistole. Type-based detection for unknown channels. In *ICASSP Proc.*, Atlanta, GA, 7–10 May 1996.

[2] M. J. Weinberger, J. J. Rissanen, and M. J. Feder. A universal finite memory source. *IEEE Trans. Info. Theory*, 41(3):643–652, May 1995.

[3] John G. Proakis. *Digital Communications*. McGraw Hill, NewYork, second edition, 1989.

[4] Owen Ernest Kelly. *Intersymbol Interference Equalization by Universal Likelihood.* PhD thesis, Rice University, ECE Dept., Houston, USA, Oct 1996.

[5] Yoram Singer. Adaptive mixtures of probabilistic transducers. 1997. To appear in Neural Computation.

[6] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens. The context tree weighting method: Basic properties. *IEEE Trans. Info. Theory*, 41(3):653–664, May 1995.

[7] Joe Suzuki. On some relationship between the context tree weighting and general model weighting techniques for tree sources. *IEEE Trans. Info. Theory*, submitted Nov. 1995.