Low Power Detection

Mohammed Nafie and Ahmed Tewfik
Dept. of Electrical Engineering, University of Minnesota
Minneapolis, MN 55455
e-mail: mnafie,tewfik@ece.umn.edu

ABSTRACT

Low power and low complexity algorithms for signal processing applications have gained increasing importance with the deployment of portable communication equipment. Most of the current power reduction techniques rely on reducing the power by VLSI implementation. This approach is expensive and limited by technology. Hence algorithm design optimization is a must for low energy consumption. Here, we propose using finite memory detection algorithms as low complexity-low energy near optimal detection algorithm that trades a small amount of detection performance for a reduction in complexity and power consumption. The negligible loss in detection performance is easily accommodated in wireless video and audio transmission applications. In data applications, this small loss can be further reduced with error correcting codes at the expense of a slight loss in communication bandwidth. We present simple algorithms for deriving the near optimum finite memory detectors in the time invariant and time variant case. The same algorithms can be used in tandem configurations in decenteralized detection.

1 Introduction

Finite memory detection has been the subject of many papers for almost three decades. It was first introduced in [1] and [2]. These two references dealt with the problem in an asymptot sense, i.e. if the number of observations is infinite. Finite memory detection can be considered as a quantized detection problem, but one in which the quantization levels at each observation are adapted based on the previous observation(s). Numerous results have been reported in the field of the theory and applications of quantized detection [3][4][5][6]. In most of that work, and due to the intractability of the probability of error function in such cases, an asymptotic measure of the performance corresponding to an infinite observation size and a very low signal to noise ratio is optimized. Since independence between observation vectors is usually assumed, the scalar quantizers for the various observation vector components were identical. Following [2], the problem of finite memory detection can be formulated as follows. After each observation,k, store only the statistic T_k. Define T_k by $T_k = f_k(T_{k-1}, X_k)$, where X_k is the current observation. The function f_k is generally time variant, but can be constrained to a time invariant function as in [2]. Also, define a decision function d_k by $d_k = d(T_k)$. If T_k can be represented by a finite number of bits, this algorithm is called finite memory detection. In [1] it was proved that using a time varying f_k you need only 2 bits to guarantee that the probability of error tends to zero as the observation size goes to infinity. Using a time invariant function $f_k = f$, it was shown in [2] that you generally don't have an optimal algorithm but rather a class of ϵ-optimal rules, i.e. for any strictly positive ϵ there is a rule, *function* f, for which the difference between the probability of error and the optimal probability of error is less than ϵ, and there is no rule for which ϵ is zero.

Minimizing the asymptotic probability of error is theoretically interesting, and perhaps even of practical interest in large sample observations, but a more important problem would be to minimize the probability of error after a finite number of observations. In [7], simple thresholding of the observation was used, with the thresholds determined by assuming an open ended problem, i.e. the number of observations is unknown. In [8] the problem of Gaussian observations was dealt with, and it was proved that simple thresholding of the observations was optimal in the time invariant case for symmetric problems if 1 bit is used to represent T_k. In [9] (and also in [10]), an optimal time varying rule was found to be a likelihood threshold rule. This simplifies the search for such a rule quite a lot. Moreover, if the likelihood ratio is monotone in the observation variable, the detector will just be a simple threshold device for the observation. But no ways for finding these thresholds were given in these papers.

The same problem, finite memory detection, was dealt with in the context of decentralized detection. Decentralized detection deals with detection problems where a center has to make the decision after receiving the observations from a set of sensors. Usually we want each sensor to transmit a quantized version of its observation, to lower the complexity or the communication requirements. If these sensors are arranged in a tandem configuration [11], then we have the same problem as in finite memory detection. For an extensive overview of that field, see [12].

This paper is organized as follows. In the next section we write the equations that need to be solved for the optimal thresholds assuming time varying thresholds. We explain the algorithms that were used to solve these equations. We then present a much simpler algorithm that yields sub-optimal thresholds and show that there is almost no difference in performance. In section 3, we briefly explain the time invariant problem. So far, we have been unable to find a simple way of obtaining the optimal thresholds. However, we present a sub-optimal method that is close to the optimal. We then present an algorithm that uses these thresholds in a low power detector. We then end by giving the conclusion.

2 Time Varying Thresholds

2.1 Equations

In this paper we will use simple thresholding of the observations as our finite memory detector. In this section we will formulate the problem of solving for the thresholds. We assume that we are using probability density functions that produce a likelihood ratio that is monotone in the observation variable. Hence the problem is reduced to thresholding the observation variable. Let us illustrate that by using a simple example. Specifically, assume that under the two hypotheses, the observations are given by $H_0 : P_0$, $H_1 : P_1$. Define the decision variable d_k as $d_k = d(T_k) \in \{H_0, H_1\}$. The probability of error after any observation can now be defined to be

$$P_k(e) = Prob.(H_1) * Prob.(d_k = H_0|H_1) + Prob.(H_0) * Prob.(d_k = H_1|H_0). \quad (1)$$

In [1], [2], the goal was to minimize the asymptotic probability of error, namely $P_\infty(e) = \lim_{k\to\infty} P_k(e)$. In this paper, and assuming we have N observations, we want to minimize $P_N(e)$. Assume that $T_k \in \{1, 2, ..., M\}$, i.e. that we can represent it by $\log_2 M$ bits. Therefore, at each observation $k \in \{2, 3, ...N-1\}$ we need to specify $M-1$ thresholds for each possible T_{k-1}, thus we need $M(M-1)$ thresholds at each observation. Notice that at the first observation we only need $M-1$ thresholds, and at the last observation, *the decision observation*, we need not calculate T_N but we only need M thresholds for calculating d_N. Take $f_{l,m}^{(k)}$ to be the l-th threshold at observation k, given $T_{k-1} = m$. Define $f_{0,m} = \infty$ and $f_{M,m} = -\infty$. Hence, the terms in the probability of error that depend on $f_{l,m}^{(k)}$ are

$$\begin{aligned}
&(Prob.(T_{k-1} = m|H_0).Prob.(f_{l-1,m}^{(k)} > X(k) > f_{l,m}^{(k)}|H_0) \\
&\quad Prob.(d_N = H_1|H_0, T_k = l) + \\
&Prob.(T_{k-1} = m|H_0).Prob.(f_{l,m}^{(k)} > X(k) > f_{l+1,m}^{(k)}|H_0) \\
&\quad Prob.(d_N = H_1|H_0, T_k = l+1)) * Prob.(H_0) \\
&+(Prob.(T_{k-1} = m|H_1).Prob.(f_{l-1,m}^{(k)} > X(k) > f_{l,m}^{(k)}|H_1) \\
&\quad Prob.(d_N = H_0|H_1, T_k = l) + \\
&Prob.(T_{k-1} = m|H_1).Prob.(f_{l,m}^{(k)} > X(k) > f_{l+1,m}^{(k)}|H_1) \\
&\quad Prob.(d_N = H_0|H_1, T_k = l+1)) * Prob.(H_1). \quad (2)
\end{aligned}$$

We then have to differentiate the above equation with respect to $f_{l,m}^{(k)}$, and set the result equal to zero to get an equation for this threshold. The problem here is that all the terms are functions of the thresholds at the other observations. Specifically, let us drop the dependence on H_i,

$$Prob.(T_{k-1} = m) = \sum_{i=1}^{M} Prob.(T_{k-2} = i) * Prob.(f_{m-1,i}^{(k)} > X(k) > f_{m,i}^{(k)}), \quad (3)$$

and assuming that d_N is equal to H_0, if the observation is larger than the threshold,

$$Prob.(d_N = H_0, T_k = l+1) = \sum_{i=1}^{M} Prob.(T_{N-1} = i) * Prob.(X(N) > f_i^{(N)}), \quad (4)$$

and the rest of the terms with d_N are calculated similarly. Therefore we have a large number of coupled equations and it is impossible to solve them directly. A way to solve them [12] would be to assume that all the thresholds are already optimal except for those at observation k. We can then get all the terms in the equation recursively using equations 3 and 4. We start the recursion with $Prob.(T_0 = i) = 1/M$ for all $i \in \{1, 2, ..., M\}$, $Prob.(T_k = l+1) = 1$ and $Prob.(T_k = i) = 0$ for all $i \in \{1, 2, ..., M\}, i \neq l+1$
Thus we can solve for the k-th observation threshold. We then repeat that process at all the observations, and iterate till we reach steady state or until the change in the thresholds is small. At each new threshold computation, the probability of error is guaranteed to decrease, and therefore we are guaranteed to arrive to a minimum. There is no guarantee that this method would go to the global minimum, but our results on Gaussian variables show that it does.

2.2 Example

Let us now illustrate the above procedure by using a simple example. Specifically, assume that under the two hypotheses, the observations are given by $H_0 : s + n(k)$, $H_1 : -s + n(k)$. Here, s is a known DC signal, and $n(k)$ is a white zero-mean Gaussian noise sequence of standard deviation σ. Differentiating (2), and setting to zero we get

$$A * P_0(f_{l,m}^{(k)}) - B * P_1(f_{l,m}^{(k)}) = 0, \quad (5)$$

where,

$$\begin{aligned}
A = Prob.(H_0) * [&Prob.(T_{k-1} = m|H_0). \\
&(Prob.(d_N = H_1|H_0, T_k = l) \\
&-Prob.(d_N = H_1|H_0, T_k = l+1))], \quad (6)
\end{aligned}$$

and

$$\begin{aligned}
B = Prob.(H_1) * [&Prob.(T_{k-1} = m|H_1). \\
&Prob.(d_N = H_0|H_1, T_k = l) \\
&-Prob.(d_N = H_0|H_1, T_k = l+1))]. \quad (7)
\end{aligned}$$

And hence, $f_{l,m}^{(k)} = \frac{\sigma^2}{2*s} \ln \frac{A}{B}$
We used the above algorithm, with 2 and 3 bits, using $s = .8$ and $\sigma = .5$. The results are compared to the optimum detector (comparing the sum of all observations with zero) in Fig. 1.

2.3 Simpler Optimization

Here, we explain a simpler optimization techniques that have less computational complexity than the above method of optimization. It has the added advantage of having a limited number of optimization variables independent on our sample size. Also in the case of on line optimization (assuming the signal value or the noise standard deviation can change on-line), this algorithm has much less complexity per sample time, as the current sample thresholds depends only on the previous thresholds and is independent of future ones. In the case of tandem decentralized detection when a center can only get information from its previous detector, this method of optimization becomes our only choice if we don't want to repeat the above optimization procedure at all the detectors. Assume that we are trying to calculate the thresholds at observation k, and we only know the optimal thresholds at all the previous observations. We apply the infinite precision optimal detector on the $M-k$ remaining observations. Assume that this detector is

$$G(X(k+1), X(k+2), ...X(M)) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma. \quad (8)$$

Starting with an initial γ, equation (2) can be written as:

$$\begin{aligned}(Prob.(T_{k-1} = m|H_0).Prob.(f^{(k)}_{l-1,m} > X(k) > f^{(k)}_{l,m}|H_0)\\ Prob.(G > \gamma|H_0, T_k = l) +\\ Prob.(T_{k-1} = m|H_0).Prob.(f^{(k)}_{l,m} > X(k) > f^{(k)}_{l+1,m}|H_0)\\ Prob.(G > \gamma|H_0, T_k = l+1)) * Prob.(H_0)\\ (Prob.(T_{k-1} = m|H_1).Prob.(f^{(k)}_{l-1,m} > X(k) > f^{(k)}_{l,m}|H_1)\\ Prob.(G < \gamma|H_1, T_k = l) +\\ Prob.(T_{k-1} = m|H_1).Prob.(f^{(k)}_{l,m} > X(k) > f^{(k)}_{l+1,m}|H_1)\\ Prob.(G < \gamma|H_1, T_k = l+1)) * Prob.(H_1). \quad (9)\end{aligned}$$

We now iterate only on $f^{(k)}_{i,m}, i \in 1, 2, ..., M-1$ and γ. We then move to observation $k+1$ and repeat the process. Here, we only make a single pass on all the observations, and hence we save a lot in the computations. Another technique for simpler optimization, which we won't discuss here, consists of dividing all future observations into a number of groups. This technique will give thresholds that are closer to the optimum thresholds, but is more complex to optimize.

3 Time Invariant Thresholds

In the time varying threshold detector, and as the number of observations increases, more memory will be needed to store all the thresholds at different observations. So, in this section, we turn our attention to time invariant detectors. There is no proof that the optimum time invariant finite memory detector is a likelihood ratio threshold device. It has only been proven in [8] that this is the case in symmetric 1-bit Gaussian detection problem. Even if that is not generally the case, this will give us lower detection power, and hence in this section we will address the problem of finding the optimum time varying thresholds. Let us assume that the thresholds are the same and equal to $f_{j,i}$ except for the first observation and the last one. Now, this is a Markov chain with the initial state given by the row vector I whose $i-th$ component is $Prob.(T_1 = i)$ and the transition matrix given by A where $A_{ij} = Prob.(f_{j-1,i} > X(k) > f_{j,i})$. Let F be a vector whose $i-th$ component is $Prob.(d_N = H_0|H_1, T_{N-1} = i)$. Therefore we can write the probability of error as $I * A^{N-2} * F$. Optimizing the thresholds in the equation is extremely difficult, and hence we have to find easier ways. A way would be to calculate the time varying thresholds and then use their average as the time invariant thresholds. We can then use the probability of error equation to optimize the initial observation and the final observation thresholds alone.

4 Low Power Detection

Assume we have already calculated the time invariant thresholds. These have to be stored and the observation quantized and compared to them. Assume we have the same hypotheses as in section 2.2. The optimal detector would be to sum all the observations and compare the sum to zero. Therefore for N observations each quantized to Q bits, we need $N * Q + 1$ one bit operations to decide between H_0 and H_1. Using a 3-bit time invariant detector we need an average number of one bit operations equal to about 44 (compared to 161) at 10 observations and 640 (compared to 1601) at 100 observations. We can reduce this even more by making the following observation: no matter what the previous comparison results were, if the final observation was higher than the maximum of all the final observation thresholds our decision would only depend on that observation. The same applies if the final observation is lower than the minimum threshold. If that is not the case, we can look at the observation previous to the final and we might be able to stop at it disregarding all the previous observations. Hence and since all the observations are identically distributed we can apply the following algorithm:

1. Compare the first observation with the M thresholds $f^{(N)}_i$, $i \in \{1, 2, ...M\}$
2. If the observation is larger than $f^{(N)}_1$ or smaller than $f^{(N)}_M$ make the decision. Else if it is between $f^{(N)}_i$ and $f^{(N)}_{i+1}$, let $T_N = i$
3. For j=N to 3
   (a) Compare observation $N-j+2$ to $f_{T_j,i}$, $i \in \{1, 2, ...M\}$
   (b) If the observation is larger than $f_{T_j,1}$ or smaller than $f_{T_j,M}$ make the decision and exit
   Else if it is between $f_{T_j,i}$ and $f_{T_j,i+1}$, let $T_{j-1} = i$
4. Compare observation N to $f^{(1)}_{T_2}$ and decide.

Using this algorithm we only need 245 one bit operations for a 3-bit time invariant detector for 100 observations.

4.1 Simulation Results

For the example in 2.2, we used the simple optimization technique to derive the thresholds. Fig. 2 shows a comparison between the probability of error performance between the thresholds obtained using this technique and the optimal thresholds. It is clear that there is no loss of performance using this simpler optimization technique. We also generated the time invariant threshold using our algorithm. An example is shown in Fig. 3. On the same figure a 2-bit detector that was derived using brute search is also shown. It is clear that our technique performs close to the optimum time invariant likelihood ratio finite memory detector.

5 Conclusion

We have presented some results in finite memory detection after proposing it being used as a low power detector. Specifically, we presented simple optimization techniques for time variant and time invariant thresholds. We also presented an algorithm that could lower the complexity even more.

References

[1] T. Cover, "Hypothesis Testing with Finite Statistics," Ann. Math. Stat., Vol. 40, pp. 828-835, 1969.

[2] M. Hellman, and T. Cover, "Learning with Finite Memory" Ann. Math. Stat., Vol. 41, No. 3, pp. 765-782, 1970.

[3] S. Kassam, "Optimum Quantization for Signal Detection," IEEE Trans. Commun., Vol. 25, No. 5, pp. 479-484, May 1977.

[4] B. Aazhang, and H. Poor, "On Optimum and Nearly Optimum Data Quantization for Signal Detection," IEEE Trans. Commun., Vol. 32, No. 7, pp.745-751, July 1984.

[5] B. Picinbono, and P Duvaut, "Optimum Quantization for Detection," IEEE Trans. Commun., Vol. 36, No. 11, pp. 1254-1258, Nov. 1988.

[6] M. Longo, T. Lookabaugh, and R. Gray, "Quantization for Decentralized Hypothesis Testing under Communication Constraints," IEEE Trans. Inform. Theory, Vol. 36, No. 2, pp.241-255, March 1990.

[7] E. Baxa, and L. Nolte, "Adaptive Signal Detection with Finite Memory," IEEE Trans. on System, Man and cybernetics, Vol. 2, No. 1, Jan. 1972.

[8] M. Freeman, "A Finite Memory, Finite Time, Gaussian Hypothesis Testing Problems," M.S. Thesis, EE Dept., MIT, Cambridge, Mass.

[9] T. Cover, M. Freeman, and M. Hellman, "OPtimal Finite Memory Learning Algorithms for the Finite Sample Problem," Information and Control, Vol. 30, pp. 49-84, 1976.

[10] R. Muise, and R. Boorstyn, "Detection with Time-varying Finite Memory Receivers," IEEE Int. Symp. on Info. Theory, pp.35-36, 1972.

[11] R. Viswanathan, S. Thomopoulos, and R. Tumuluri, "Optimal Serial Distributed Decision Fusion," IEEE Trans. on Aerospace and Electronic Systems, Vol. 24, pp. 366-375, July 1988.

[12] J. Tsitsiklis, "Decentralized Detection," Advances in Statistical Signal Processing, Vol. 2, pp. 297-344, JAI Press, 1993.

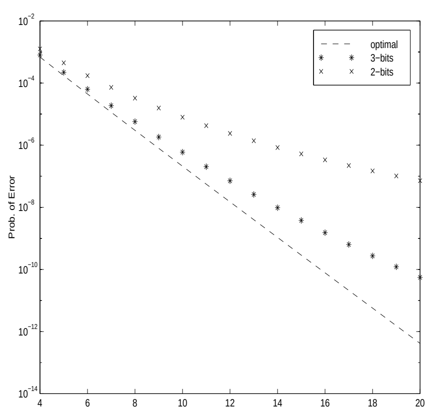

Figure 1: Optimal Vs. Finite Memory Detector

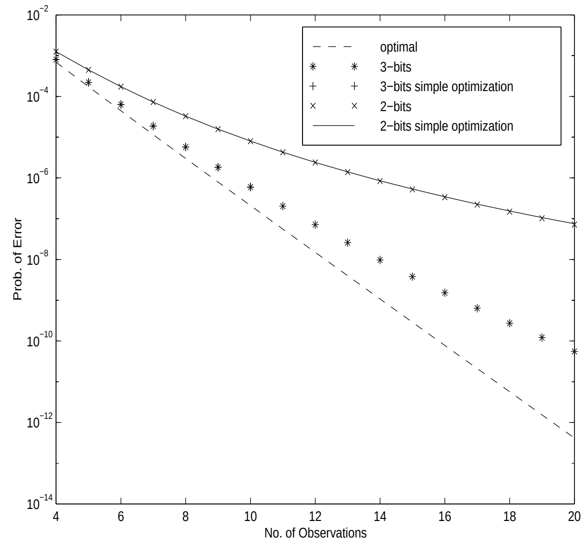

Figure 2: Comparing Optimization Techniques

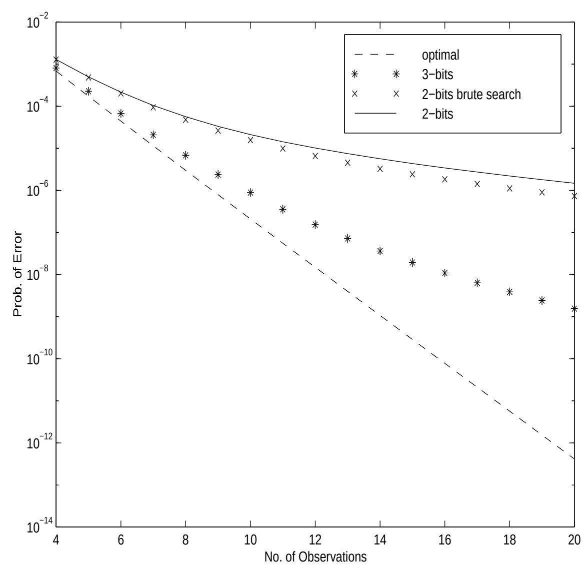

Figure 3: Time Invariant Detector