E6712: Homework 1 - Solution

Prepared by Anmo Kim and Emina Soljanin

1. Consider the following binary detection problem discussed in class:

$$r = s_i + n, \quad i \in \{1, 2\},$$

where n is an N-dimensional noise vector whose components are independent identically distributed Gaussian random variables with zero mean and unit variance, and the two possible signals are the following N-dimensional vectors:

$$s_1 = (m, m, \ldots, m) \text{ and } s_2 = (0, 0, \ldots 0).$$

Assume that the prior probabilities of the two messages are equal.

(a) Evaluate analytically the error probability performance of an optimal receiver for $N = 1$ and $m \in \{1, 2, 3, 4, 5, 6\}$, and for $N = 4$ and $m \in \{1, 2, 3, 4, 5, 6\}$. Plot the error probability v.s. m on a single graph for both $N = 1$ and $N = 4$. using the logarithmic scale for the error probability axis.

(b) Write a computer program (see the note below) to evaluate by simulation the error probability performance for the same parameters as in a). Plot the results on the same graph as for a) for comparison.

(c) Compute the Chernoff bound on the probability of error evaluated analytically in a) and by simulation in b), and plot on the same graph as for a) and b).

Solution

We have derived in class that

$$P(\text{error}) = Q\left(\frac{\sqrt{\sum_j (s_{2j} - s_{1j})^2}}{2\sigma}\right).$$

For $s_1 = (m, m, \ldots, m)$ and $s_2 = (0, 0, \ldots 0)$, we get

$$P(\text{error}) = Q\Big(\frac{\sqrt{N}m}{2}\Big)$$

The Chernoff bound says that

$$Q(x) \le \exp\Big(-\frac{x^2}{2}\Big),$$

and thus, for our problem,

$$P(\text{error}) = Q\Big(\frac{\sqrt{N}m}{2}\Big) \le \exp\Big(-\frac{m^2 N}{8}\Big).$$

With Matlab or C, we can calculate the Q function by using the erf function which has the following relation:

$$Q(x) = \frac{1}{2} \cdot \Big[1 - \text{erf}\Big(\frac{x}{\sqrt{2}}\Big)\Big]$$

The following is the definition of erf function in Matlab. (We can see this definition by typing "`help erf`" on the Matlab command prompt. The "`help`" command can be used for any other Matlab functions.)

```
>> help erf

 ERF Error function.
    Y = ERF(X) is the error function for each element of X.  X must be
    real. The error function is defined as:

      erf(x) = 2/sqrt(pi) * integral from 0 to x of exp(-t^2) dt.
```

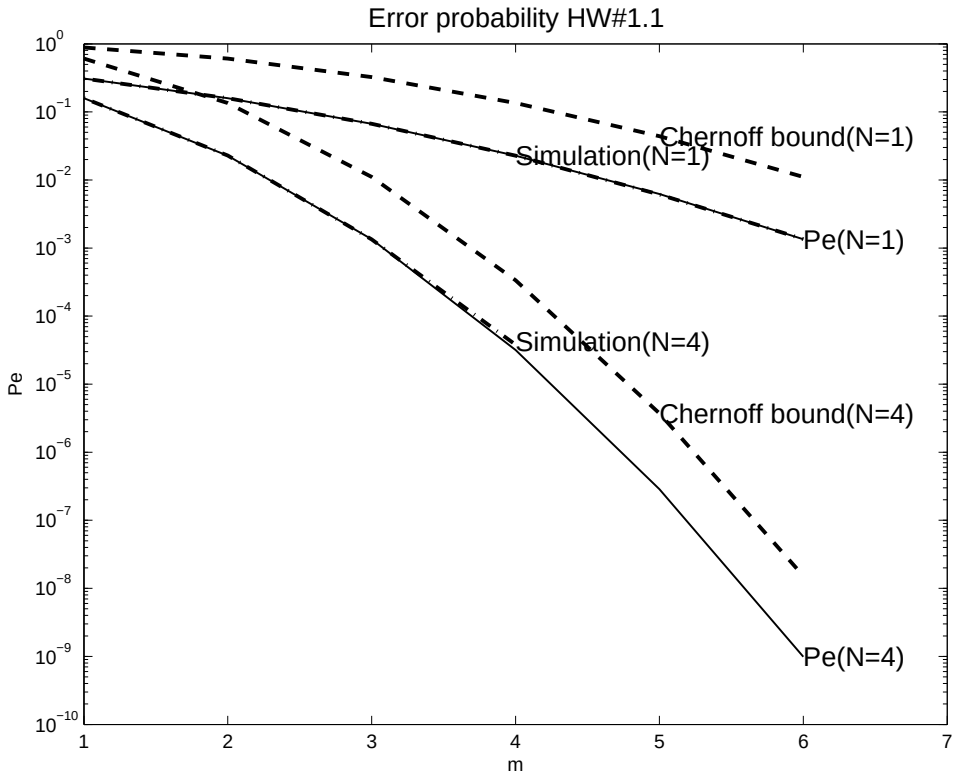
For the Matlab beginners, "`lookfor`" command would be also useful. The following is the help document about the `lookfor` command.

```
>> help lookfor

 LOOKFOR Search all M-files for keyword.
    LOOKFOR XYZ looks for the string XYZ in the first comment line
    (the H1 line) of the HELP text in all M-files found on MATLABPATH.
    For all files in which a match occurs, LOOKFOR displays the H1 line.

    For example, "lookfor inverse" finds at least a dozen matches,
    including the H1 lines containing "inverse hyperbolic cosine"
    "two-dimensional inverse FFT", and "pseudoinverse".
    Contrast this with "which inverse" or "what inverse", which run
    more quickly, but which probably fail to find anything because
    MATLAB does not ordinarily have a function "inverse".
```

The following graph shows all plots for the problem.1:



The Matlab code for the above result will be provided at the end of this solution.

2. Consider the same signaling scheme as in Problem 1) for arbitrary N, but with a suboptimal receiver which operates as follows. After each of the N symbol intervals, the receiver decides whether an m or a 0 was sent according to the rule that is optimal for $N = 1$. After the N symbol intervals, the receivers declares message s_1 if the majority of its per-symbol decisions were m, and message s_2 if the majority of its per-symbol decisions were 0. If there was a tie in the per-symbol decisions (in the case of an even N), the receiver decides either s_1 or s_2, each with probability 1/2.

   (a) Derive an expression for this receiver's error probability, and its Chernoff bound.

   (b) Write a computer program to evaluate by simulation the error probability performance for $N = 4$ and $m \in \{1, 2, 3, 4, 5, 6\}$. Evaluate analytically the error probability performance for the same parameters based on the expression derived in a). Compute the Chernoff bound. Plot the error probability (simulated, computed, and the Chernoff bound) v.s. m on a single graph.

   (c) How does the performance of this receiver differ from the performance of the optimal receiver?

Solution

(a) The receiver makes a symbol error with probability

$$p = Q\left(\frac{m}{2}\right).$$

For odd N, the probability of message error is

$$P(\text{error}) = \sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} p^i (1-p)^{N-i}.$$

For even N, the probability of message error is

$$P(\text{error}) = \sum_{i=1+N/2}^{N} \binom{N}{i} p^i (1-p)^{N-i} + \frac{1}{2}\binom{N}{N/2}[p(1-p)]^{N/2}.$$

The general Chernoff bound derived in class is

$$P(X \geq \varepsilon) \leq e^{-\varepsilon s} E[e^{sX}].$$

For the Binomially distributed random variable X, we have

$$\begin{aligned} E[e^{sX}] &= \sum_{i=1}^{N} \binom{N}{i} e^{si} p^i (1-p)^{N-i} \\ &= [e^s p + (1-p)]^N, \end{aligned}$$

and thus

$$P(X \geq \varepsilon) \leq e^{-\varepsilon s}[e^s p + (1-p)]^N.$$

Let the right hand side equation be $g(s)$. Then, $g(s)$ can be minimized over s by differentiation:

$$\begin{aligned} \frac{dg(s)}{ds} =& -\varepsilon e^{-\varepsilon s}[e^s p + (1-p)]^N + N e^{-\varepsilon s}[e^s p + (1-p)]^{(N-1)} e^s p \\ =& e^{-\varepsilon s}[e^s p + (1-p)]^{N-1}\Big[N e^s p - \varepsilon(e^s p + (1-p))\Big] \end{aligned} \tag{1}$$

Now,

$$\begin{aligned} \frac{dg(s)}{ds} = 0 &\Leftrightarrow \\ -\varepsilon[e^s p + (1-p)] + N e^s p = 0 &\Leftrightarrow \\ (N-\varepsilon) p e^s = \varepsilon(1-p) &\Leftrightarrow \\ e^s = \boxed{z_{\min} \triangleq \frac{\varepsilon}{N-\varepsilon}\frac{1-p}{p}} & \end{aligned}$$

From (1), we see that

$$\frac{dg(s)}{ds} = \begin{cases} < 0, & e^s < z_{\min}; \\ = 0, & e^s = z_{\min}; \\ > 0, & e^s > z_{\min}. \end{cases}$$

Therefore $g(s)$ has a minimum at $e^s = z_{\min}$.

The Chernoff bound was derived under the assumption that $s > 0$. Therefore, we should check if $z_{\min} > 1$. This definitely holds for our case of interest: $\varepsilon > N/2$ and $p < 1/2$.

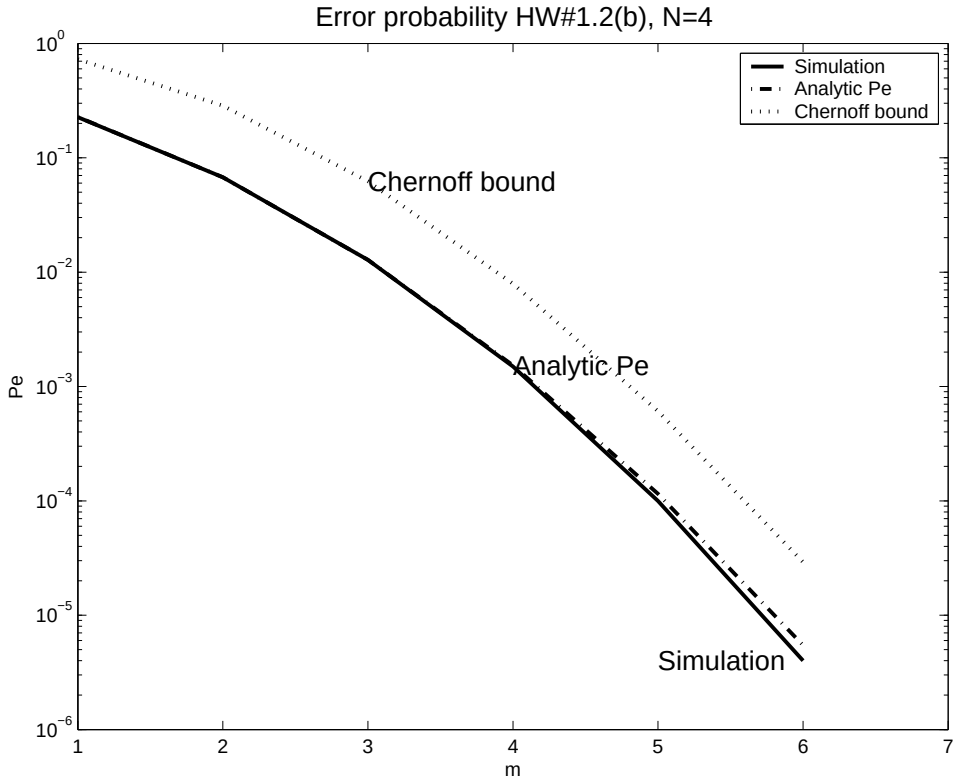
Finally, the Chernoff bound for a Binomially distributed random variable X is

$$P(X \geq \varepsilon) \leq \Big[\frac{1-p}{p}\frac{\varepsilon}{N-\varepsilon}\Big]^{-\varepsilon}\Big[(1-p)\frac{N}{N-\varepsilon}\Big]^{N}$$

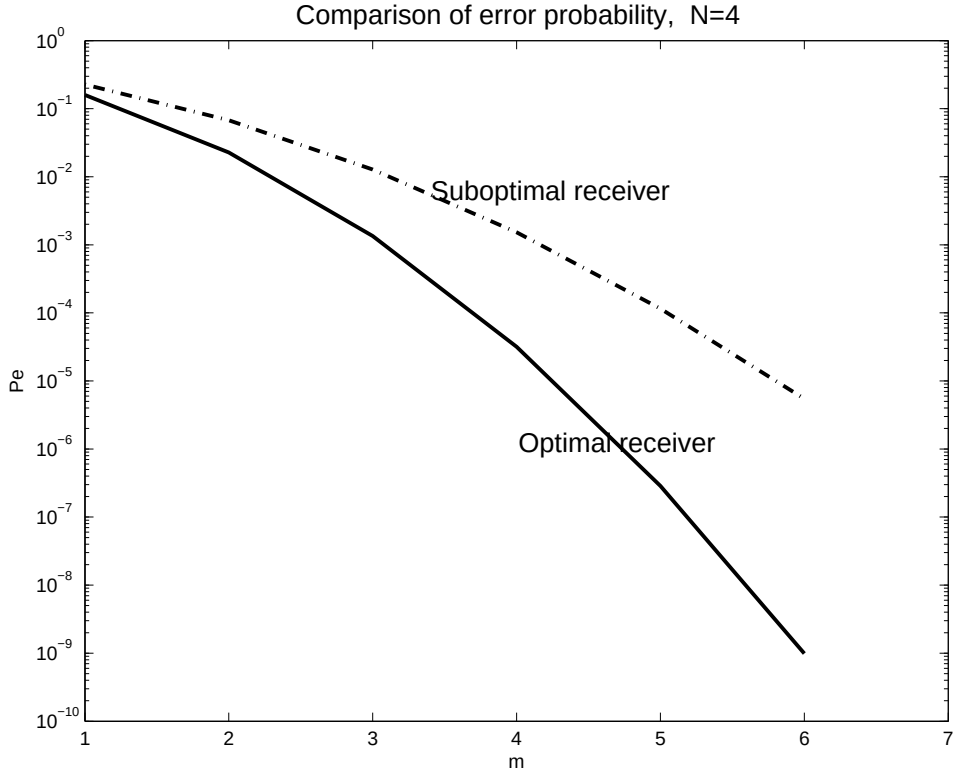
To bound the P(error) for both even and odd N, we can take $\varepsilon = N/2$, which gives

$$P(\text{error}) \leq \big[2\sqrt{p(1-p)}\big]^{N}.$$

(b) The plot will be drawn as follows:



(c) The following graph shows two error probability lines, one for the optimal receiver and the other for the suboptimal receiver. We can easily find that optimal receiver shows quite low error probability.



3. On-Off Keying (OOK) is a modulation scheme suitable for the optical direct detection channel. Under OOK, information is conveyed by turning the laser on for T seconds to send the symbol 1 and turning the laser off for T seconds to send the symbol 0. The information available at the receiver is the number of photons observed in a symbol interval. This number is modelled as a Poisson random variable with mean $(\lambda_s + \lambda_n)T$ when 1 is sent and $\lambda_n T$ when 0 is sent.

   (a) Derive the rule on which an optimal receiver for this channel should base its decision.

   (b) Derive an expression for the exact error-probability of this receiver.

Solution

(a) Let K denote the number of photons observed by the receiver. The optimal detector rule is

$$\frac{P(K/1 \text{ sent}) \cdot P(1)}{P(K/0 \text{ sent}) \cdot P(0)} \underset{0}{\overset{1}{\gtrless}} 1$$

Since

$$P(K/1 \text{ sent}) = \frac{[(\lambda_s + \lambda_n)T]^K}{K!} e^{-(\lambda_s+\lambda_n)T} \text{ and } P(K/0 \text{ sent}) = \frac{(\lambda_n T)^K}{K!} e^{-\lambda_n T},$$

$$\frac{P(K/1 \text{ sent}) \cdot P(1)}{P(K/0 \text{ sent}) \cdot P(0)} = \Big(\frac{\lambda_s + \lambda_n}{\lambda_n}\Big)^K e^{-\lambda_s T} \cdot \frac{P(1)}{P(0)}$$

Taking ln, we get

$$K \underset{0}{\overset{1}{\gtrless}} \frac{\lambda_s T + \ln\Big(\frac{P(1)}{P(0)}\Big)}{\ln\Big(1 + \frac{\lambda_s}{\lambda_n}\Big)} = \gamma$$

(b) The error-probability depends on γ.

$$\begin{aligned} P(\text{error}) &= P(\text{error},\ 0 \text{ sent}) + P(\text{error},\ 1 \text{ sent}) \\ &= P(0) \cdot P(\text{error}/0 \text{ sent}) + P(1) \cdot P(\text{error}/1 \text{ sent}) \\ &= P(0) \cdot P(K > \lfloor \gamma \rfloor) + P(1) \cdot P(K \leq \lfloor \gamma \rfloor) \\ &= P(0) \cdot \sum_{j=\lfloor \gamma \rfloor + 1}^{\infty} \frac{(\lambda_n T)^j}{j!} \cdot e^{\lambda_n T} + P(1) \cdot \sum_{j=0}^{\lfloor \gamma \rfloor} \frac{((\lambda_s + \lambda_n) T)^j}{j!} \cdot e^{(\lambda_s + \lambda_n) T} \end{aligned}$$

Matlab Code

- Q-function using erf function

```
%filename: Qfunction.m
function res = Qfunction(x) res=1/2*(1-erf(x/sqrt(2)));
```

- Problem.1

```
%hw1_1.m
%Anmo Kim

m=1:6;

%calculate error probability of optimal receiver for Gaussiand channel
%using Q-function ============================================================
Pe1=Qfunction(m/2); %error when N=1
Pe4=Qfunction(m); %error when N=4

%Chernoff bound  =============================================================
chern1 = exp(-m.^2./8); %N=1
chern4 = exp(-m.^2./2); %N=4

%Simulation      =============================================================
n_experiment=1000000; %number of experiments

% N=1
for k=1:m(length(m))
   mk = m(k);
   s=mk*floor(2.*rand(n_experiment,1)); % binary sequence with p(0)=p(1)=1/2
   n=randn(n_experiment,1); % noise generation
   r=s+n;   %received signal which has gone through additive channel
   thr=mk/2;  %threshold m/2 is optimal for unipolar signal
   r=r-ones(n_experiment,1)*thr;
   r=sign(sign(r)+1); % extracting binary sequence
   r=abs(r-s/mk); %compare with the signal sent originally
   Re1(mk)=sum(r)/n_experiment; %error rate
end

%N=4;
for k=1:m(length(m))
    mk=m(k);
   s=mk*floor(2.*rand(n_experiment,1));
   n=randn(n_experiment,4);
   r=s*ones(1,4)+n; %signal after experiencing the AWGN channel
```

```
   thr=mk*4/2;

   r(:,1)=sum(r,2);
   r(:,1)=r(:,1)-ones(n_experiment,1)*thr;
   r(:,1)=sign(sign(r(:,1))+1);
   r(:,1)=abs(r(:,1)-s/mk);
   Re4(mk)=sum(r(:,1))/n_experiment; %error rate
end
%end of simulation ======================================================

%plotting ===============================================================
semilogy(m, Pe1, 'k-', m, Pe4, 'k-', m, chern1, 'k--', m, ...
    chern4, 'k--', m,Re1,'k-.',m,Re4,'k-.');

axis(axis+[0 +1 0 0]);
xlabel('m');
ylabel('Pe');
title('Error probability HW#1.1');
%end of hw1_1.m
```

- Problem.2(b)

```
%hw1_2b.m
%Anmo Kim
warning off MATLAB:conversionToLogical;

%Defining some constants
m=1:6; N=4; Pa=0; Pc=0; n_experiment=1000000;

%Simulation=============================================================
for k=1:m(length(m))
   s=k*floor(2*rand(n_experiment,1));
   n=randn(n_experiment,N);
   r=s*ones(1,N)+n; %signal after experiencing the AWGN channel
   thr=k/2;         %threshold

   %detection
   r=r-thr;
   r=sign(sign(r)+1);
      %Here, If gaussian noise is exactly zero, then recognized as "1"
   r(:,1)=sign(sum(r,2)-(N/2));
   coin=~logical(r(:,1)).*floor(2*rand(n_experiment,1));
      %If number of 1 and 0 is same in r, choose 1 or 0 randomly with p=.5
   r(:,1)=(r(:,1)-coin);
   r(:,1)=sign(r(:,1)+1);

   %error check
   Ps(k)=sum(abs(r(:,1)-s/k))/n_experiment;
end


%Analytical P(error) and Chernoff bound of it=======================
p = Qfunction(m/2); %Analytical error probability
```

```
for k = ceil((N+1)/2):N Pa = Pa +
nchoosek(N,k)*(p.^k).*((1-p).^(N-k));
   %Analytical probability of error
end

if (ceil(N/2)==N/2) %if N is even number
    Pa = Pa + 1/2*nchoosek(N, N/2)*(p.^(N/2)).*((1-p).^(N/2));
end

Pc = (2*sqrt(p.*(1-p))).^N;
   %Chernoff bound for the probability of error


%plotting=====================================================
semilogy(m, Ps, 'k-', m, Pa, 'k-.', m, Pc, 'k:');
axis(axis+[0 +1
0 0]);
xlabel('m');
ylabel('Pe');
title('Error probability HW#1.2(b), N=4');
text(m(length(m)-1), Ps(m(length(m))), 'Simulation');
text(m(length(m)-2), Pa(m(length(m))-2), 'Analytic Pe');
text(m(length(m)-3), Pc(m(length(m))-3), 'Chernoff bound');
legend('Simulation', 'Analytic Pe', 'Chernoff bound',1);
%end of hw1_2b.m
```

- Problem.2(c)

```
%hw1_2c.m
%Anmo Kim
%Comparison of optimal and suboptimal receiver

m=1:6; N=4; Pa=0;

%error probability of the optimal receiver -- Problem.1
Pe4=Qfunction(m); %error when N=4

%error probability of the suboptimal receiver -- Problem.2
p = Qfunction(m/2); %Analytical error probability

for k = ceil((N+1)/2):N Pa = Pa +
nchoosek(N,k)*(p.^k).*((1-p).^(N-k));
   %Analytical probability of error
end

if (ceil(N/2)==N/2) %if N is even number
    Pa = Pa + 1/2*nchoosek(N, N/2)*(p.^(N/2)).*((1-p).^(N/2));
end

%plotting
semilogy(m, Pe4, 'k-', m, Pa, 'k-.');
axis(axis+[0 +1 0 0]);
xlabel('m');
ylabel('Pe');
```

```
title('Comparison of error probability, N=4');
text(m(length(m)-2), Pe4(m(length(m))-2), 'Optimal receiver');
text(m(length(m)-2), Pa(m(length(m))-2), 'Suboptimal receiver');
%end of hw1_2c.m
```

–end of HW#1 solution