Homework Help - Problem Set 2
Solutions

[Bias and error probability help] Appendix D addresses Mean-Square Error (MSE), but basically as the name says, it is the mean of the difference between the two quantities, namely the error, squared. Thus if

$$a = b + e$$

and the error between a and b is $e = a - b$, then the MSE is simply $\mathbb{E}[e^2]$ if e is AWGN. This applies to the AWGN when $a = \boldsymbol{y}$ and $b = H \cdot \boldsymbol{x}$ and the error is the AWGN noise $\boldsymbol{n}$. The reverse direction of estimating $\boldsymbol{x}$ from $\boldsymbol{y}$, however, is closer to the detection problem in digital transmission. This is especially true if $\boldsymbol{x}$ is a discrete random vector. The average error probability is well approximated by

$$P_e \approx N_e \cdot Q(\frac{d_{min}}{2\sigma})$$

where d_{min} just computes the the difference between the the corresponding noise-free channel outputs, so

$$d_{min} = \min_{\boldsymbol{x}' \neq \boldsymbol{x} \in C_{\boldsymbol{x}}} \|H \cdot \boldsymbol{x} - H \cdot \boldsymbol{x}'\| \ .$$

When $H = I$, this is simply the distance between the closest two code words. The quantity N_e is the average number of nearest neighbors, which in most cases is closely approximated by the number of other codewords that may be at the minimum distance. Chapter 1 addresses this area if more information required (but this is sufficient for the homework). A binary constellation simply has d_{min} as the distance between the two points and $N_e = 1$. A 4PAM constellation $\pm 1, \pm 3$ has distance $d_{min} = 2$ and an average of 1.5 nearest neighbors for each point, so $P_e \approx 1.5Q(1/\sigma)$ (the approximation is exact in this example). So if the 4PAM transmission system had $\bar{\mathcal{E}}_x = 5$ and $\sigma^2 = .04$ with $H = 1$, then a 4PAM system would have $d/(2 \cdot \sigma) = 2/(2 \cdot .2) = 5$, and the symbol error probability is then $P_e = 1.5 \cdot Q(5)$; in matlab:

```
>> 1.5*Q(5) =   4.2998e-07
```

The linear MMSE estimator of channel input, given channel input is always biased with non-zero noise. Even in the simplest case of $y = x + n$, it is possible to multiply y by a number less than 1 that will have a MMSE that reduces the signal x just enough so that the consequent simultaneous decrease of noise is beneficial. If the data signal has energy such that $SNR = \bar{\mathcal{E}}_x/\sigma^2$, this shrinkage of the channel output produces $(1 - 1/SNR) \cdot x_k$ and a ratio $\bar{\mathcal{E}}_x/MMSE = SNR + 1$. That apparently larger SNR misleads somewhat in that the estimate is biased. Bias removal (multiply by $(SNR + 1)/SNR$ eliminates the bias and increases the noise back to the original level. Clearly, the MMSE would not help the $x + n$ channel, but with more general channels the MMSE estimate can reduce substantially crosstalk and intersymbol interference; however the same bias/scaling-error occurs. This amount remains $SNR_{mmse} = \bar{\mathcal{E}}_x/M\bar{M}SE$ (with $SNR = SNR_{mmse} - 1$ and the same scale-up removes the bias.

A detector designed for the original x has a non-zero-mean Gaussian noise, given the input x. A non-zero (conditional) mean causes the error-probability calculation to reduce the minimum distance by the bias amount. The 4 PAM example above has $SNR = 25$ (14 dB). The bias then causes

$$d_{min,bias} \to d_{min} \cdot (1 - 1/25) = \frac{24}{25} \cdot d_{min}$$

or equivalently the $d_{min}/2$ in the P_e formula's Q-function argument subtracts $d_{min} \cdot 1/25$ while the $\sqrt{MMSE}$ reduces to $\sqrt{24/25} \cdot \sigma$. The numerator reduction offsets the denominator reduction in the trivial scalar AWGN case. However, if there had been residual ISI added to the noise at this MMSE biased SNR of 25 dB, the minimum distance decreases by the same amount in bias removal, but the ISI would reduce presumably by a larger amount than the noise, and the improvement in distance to residual error would more than offset. So, removing the bias produces a better (lower) error probability, here corresponding to an SNR of 24, which still exceeds the value ocuring if some other non-MMSE estimate of x is instead used.. This holds in general - always remove the bias, it always improves. In more general cases, despite the slight reduction from $SNR_{mmse} \to SNR_{mmse} - 1$ to remove the bias, this new MMSE-assisted receiver will always perform at least as well as if there were no MMSE estimator.

[**Matrix AWGN MMSE**] A matrix channel example appears at the end of Lecture 3. It is a simple MMSE example for matrix AWGN. The channel is just 2×2, but uses the basic formula $W = R_{\boldsymbol{x}\boldsymbol{y}} \cdot R_{\boldsymbol{y}\boldsymbol{y}}^{-1}$. For matrix AWGN channels H, $R_{\boldsymbol{x}\boldsymbol{y}} = R_{\boldsymbol{x}\boldsymbol{x}} \cdot H^*$ and $R_{\boldsymbol{y}\boldsymbol{y}} = H \cdot R_{\boldsymbol{x}\boldsymbol{x}} \cdot H^* + R_{\boldsymbol{n}\boldsymbol{n}}$. The MMSE then proceeds directly in Matlab.

```
>>H = 1.0000    0.4000
            0.2000    0.1000
>> Rxx=eye(2);
>> Ryy=H*Rxx*H'+Rnn;
>> Ryx=H;
>> W=(Ryx')*inv(Ryy) =
    0.9524   -0.4762
    0.0000    1.6667
>> W*H = 0.8571    0.3333
```

```
                   0.3333    0.1667
>> Ree=Rxx-W*Ryx =
    0.1429   -0.3333
   -0.3333    0.8333
>> snr=det(Rxx)/det(Ree) =  126.0000
>> b=log2(snr)  =    6.9773
>> SNR=inv(diag(diag(Ree))) =
7.0000         0
     0    1.2000

>> log2(diag(SNR)) =
    2.8074
    0.2630
>> sum(log2(diag(SNR)))  =    3.0704
```

The bias could be removed, but the matlab commands recognize that the formula for bit rate has $1 + SNR$ in it already, which is directly the biased SNR; so the implementation removes the bias for the detector, but the analysis need not address it.

[**Entropy Help**] Section 2.3 describes entropy and differential entropy. However, a confusion point generally (beyond the class homework assignments) can be the normalization. The confusion often arises from the correct dimensionality. For a symbol (codeword) $\boldsymbol{x}$ with N real dimensions, the entropy uses the full probability distribution $p_{\boldsymbol{x}}$. Each codeword has a probability; often at this symbol level they are all equally likely, but that need not always hold. When equally likely, the entropy simply is $\log_2(2^b) = b$, and so equal to the number of bits/symbol. This uniform distribution provides highest entropy.

However, marginal distributions for subsymbols may not have equally likely points – even if all the full-length codewords are equally likely. The classic example of this is an infinite-dimensional set of codewords with equally likely occurrence and uniform spacing throughout a hypersphere has marginal distributions in any finite set of dimensions that approach Gaussian. Thus, the entropy of the subsymbol $\tilde{\boldsymbol{x}}$ is $\mathcal{H}_{\tilde{\boldsymbol{x}}}$ and computes entropy for the marginal distribution that applies to the subsymbol (presumably the same in all subsymbols, but often not uniform for the subsymbol points). The different subsymbol probabilities allow for redudancy and code design over the entire symbol length. Sequences of subsymbols can separate more effectively (increasing minimum distance and overall performance) if the marginal distributions do not use all the points, and sometimes this leads to non-uniform distributions. The entropy formula otherwise applies, but to the marginal distribution $p_{\tilde{\boldsymbol{x}}}$. There are two possibly different entropies with tilde's:

$$\mathcal{H}_{\tilde{\boldsymbol{x}}} = \mathbb{E}\left[\log_2\left(p_{\tilde{\boldsymbol{x}}}\right)\right]$$

and

$$\widetilde{\mathcal{H}}_{\boldsymbol{x}} = \frac{\mathcal{H}_{\boldsymbol{x}}}{\overline{N}}$$

where again $\overline{N} = N/\widetilde{N}$. Similarly, $\mathcal{H}_x$ and $\overline{\mathcal{H}}_{\boldsymbol{x}}$

$$\mathcal{H}_x = \mathbb{E}\left[\log_2\left(p_x\right)\right]$$

and

$$\overline{\mathcal{H}}_{\boldsymbol{x}} = \frac{\mathcal{H}_{\boldsymbol{x}}}{N} \ .$$

Differential entropy's maximizing $\mathcal{H}_x$ distribution (with an energy constraint) is Gaussian, so not uniform. However, this is because differential entropy tacitly presumes an asymptotic equipartition situation where the continuous distribution arises through random code selection (and thus random subsymbol selection) from a uniform distribution (which is maximum entropy $\mathcal{H}_{\boldsymbol{x}}$ for the energy-constrained infinite-dimensional hypersphere). There is really no $\bar{\mathcal{H}}_{\boldsymbol{x}}$ with infinite dimensions, but really only $\mathcal{H}_x$ that has useful meaning. However, it could be argued that $\bar{\mathcal{H}}_{\boldsymbol{x}} = \mathcal{H}_x$ in this case, so equality can occur but is not necessarily always applicable.

MMSE and Entropy Conditional entropy and mutual information essentially generalize the Gaussian channel's MMSE and SNR to general probability distributions. Given two random variable/vector processes $\boldsymbol{x}$ and $\boldsymbol{y}$ with joint probability distribution, the problem of estimating the common part between them is essentially communication. One direction (e.g., matrix AWGN) is the forward direction, while the other reverse direction is the backward direction. The SNR's (biased and unbiased) are the same for both directions because the mutual information is symmetric; however the MSE's and energies are not the same for forward and backward, must their ratios. Some easy expressions follow when the joint distribution is Gaussian.

The forward direction $\boldsymbol{y}H \cdot \boldsymbol{x} + \boldsymbol{n}$ (with white noise) thus has estimator $H = R_{\boldsymbol{y}\boldsymbol{x}} \cdot R_{\boldsymbol{x}\boldsymbol{x}}^{-1}$, so forms easily once the two matrices are known. These either come from the Gaussian probability distribution directly, or can be computed readily from a given channel as in the notes. The backward direction similarly is $\boldsymbol{x} = W \cdot \boldsymbol{y} + \boldsymbol{e}$, with error autocorrelation matrix $R_{\boldsymbol{e}\boldsymbol{e}}$ and $W = R_{\boldsymbol{x}\boldsymbol{y}} \cdot R_{\boldsymbol{y}\boldsymbol{y}}^{-1}$. The symmetry of $\boldsymbol{x}$ and $\boldsymbol{y}$ in the expressions suggests the equal common part - SNR or mutual information. Finding these matrices and MMSE error autocorrelation matrices appeared earlier in this HWH (PS2.3).

For real-baseband, the conditional entropy basically has $1/2\log_2$ of the constant $(2\pi e)^N$, where N is the number of real (error vector) dimensions, times the determinant of the error's autocorrelation matrix. The determinant (which is product of eigvenvalues) should discard any zero eigenvalues. The entropy can be normalized to the dimensionality of either vector, but by digitial-transmission convention $\boldsymbol{x}$ (channel input) is chosen. An interesting expression (see Appendix D) is that for square matrices

$$\mathcal{I} = log_2 \left|I - W \cdot H\right|^{-1} = log_2 \left|I - H \cdot W\right|^{-1} \quad .$$