Homework Help - Problem Set 2
Solutions

[Bias and error probability help (4.29)]

This problem is again review of bias to refresh your 379A recollection. It has an interesting twist in that there are 3 messages (so a "trit").

a. This forward-channel MMSE is just the noise itself.

b. There are 3 points, so just find Pe for each and multiply by neighbors. $N_e > 1$ for 3 levels.

c. This is a very easy 2nd derivative so if you are getting something more complex, restart.

d. Just need to plug in the α value in your MMSE from Part c.

e. The decision boundary is incorrect, so the distance between points is affected and the P_e found increases consequebntly.

f. Fixing the boundaries is another way that is equivalent to removing the bias through scaling. This last part illustrates the scaling of the decision boundary, instead MMSE filter output, to remove bias.

[Spatial Equalizer (4.36)]

This problem follows the last problem on PS1 (4.25), which you by now have completed and have solutions. Thus, a little review of that can help ease this problem.

a. Recall the energy is the trace of the autocorrelation matrix.

b. Recall $\bar{R}_{xy}(D) \stackrel{\Delta}{=} \mathbb{E}\left[\boldsymbol{x} \cdot \boldsymbol{y}\right]$.

c. Recall the orthognality principle or just use the formula for the MMSE estimator, $R_{\boldsymbol{x}\boldsymbol{y}} \cdot R_{\boldsymbol{y}\boldsymbol{y}}^{-1}$.

d. Similar use of MMSE formula for $R_{\boldsymbol{e}\boldsymbol{e}}$ and compute its trace.

e. Look at diagonal elements of $W \cdot H$.

f. MMSE is only optimum for a matrix channel if it is vector coding.

General Entropy and MMSE review: 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

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

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 (2.10)**] This is a refresher problem on dimenstionality as well as cultivates facility between constellation/codes and entropy. The unequally likely points in subsymbols often associate whit shaping gain, but here we associate them also with redundancy and entropy.

a. Remember no (outer, outer) can occur, and indeed the points in the 2D subsymbols are not equally likely, even though the 4D points are equally likely.

b. If Part a is correct, this answer is its reciprocal.

c. This entropy has tildes and bars on the $\mathcal{H}$'s, not the $\boldsymbol{x}$'s. It is relatively easy.

d. This entropy now works with a subsymbol constellation where points are not equally likely. There are only two possibly probability values, so find those two values and multiply the log of each's inverse by its frequency of occurrence, and sum. These entropies should exceed the code redundancy in order for the code to be feasible to implement, and of course they do here.

e. There are only three possibly probability values, so find those three values and multiply the log of each's inverse by its frequency of occurrence, and sum. Similar to previous part.

f. The maximum entropies of course should be higher than all the rest in this problem.

Capacity in bits/second (2.14)

a. This should be easy now as application of $0.5 \cdot \log_2(1 + SNR)$.

b. Multiply Part a answer by number of dimensions/second.

c. Remember trade of factor $a > 1$ in bandwidth for factor $1/a$ in energy is always a good trade on the AWGN.

d. Insert $a = 100$ into Part c answer.

e. Can we always get a factor of 100x increase in bandwidth in practice?

MMSE Estimation (2.20)

a. This is the right-side matrix dimension.

b. This is simple matlab exercise on $W = R_{\boldsymbol{x}\boldsymbol{y}} \cdot R^{-1}_{\boldsymbol{y}\boldsymbol{y}} = H^* \left[H \cdot H^* + I\right]^{-1}$.

c. The forward channel estimator is H itself (with white noise).

d. Just looking for name here.

e. We look for two names here that are equivalent as $N \to \infty$.

f. Use the matrix formula $\widetilde{\mathscr{H}}_{\boldsymbol{x}/\boldsymbol{y}} = \frac{1}{2}\log_2(4\pi^2 e^2 \left|R_{\boldsymbol{e}\boldsymbol{e}}\right|)$ and use it again flipping x and y.

g. This SNR should be the same in both forward and backward forms, because the mutual information is symmetric.

h. This is basically restarting Chapter 2"s sphere-packing results.