Proceedings of the 2000 IEEE Workshop on Speech Coding
September 17-20, 2000, Delavan, Wisconsin, USA

RESULTS ON REVERSE WATER-FILLING, SNR, AND LOG-SPECTRAL ERROR IN CODEBOOK-BASED CODING

Stephen Voran
Institute for Telecommunication Sciences
sv@its.bldrdoc.gov

ABSTRACT

This paper identifies optimum levels of reverse water-filling for codebook-based coding of noise and speech signals. We find that there is little to be gained from optimizing an effective rate parameter. We identify trade-offs between SNR and log-spectral error. We show that the use of a gain factor compares favorably with reverse water-filling in some situations.

1. BACKGROUND

Let $\boldsymbol{x}$ be an input signal vector provided to a coder, and let $\hat{\boldsymbol{x}}$ be the corresponding output signal vector. Define the error signal vector $\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}}$. The power spectral densities (PSD's) of these three signals are related by

$$P_{ee}(\omega) = P_{xx}(\omega) - 2\,re(P_{e\hat{x}}(\omega)) - P_{\hat{x}\hat{x}}(\omega). \qquad (1)$$

[1] and [2] show that when stationary Gaussian signals are coded at low bit rates and that coding is near-optimum with respect to a squared-error distortion measure, then the relationship between the power spectral density of the coded signal $\hat{\boldsymbol{x}}$ and the PSD of the original signal $\boldsymbol{x}$ is

$$P_{\hat{x}\hat{x}}(\omega) = \max\left[P_{xx}(\omega) - D(R),\ 0\right]. \qquad (2)$$

This result would follow from (1) when $\boldsymbol{e}$ and $\hat{\boldsymbol{x}}$ are uncorrelated and when $P_{ee}(\omega)$ is replaced with *D(R)*, the frequency-independent value of coding distortion for a coder operating at *R* bits/sample. From [1] we know that for a zero-mean white Gaussian source with variance σ_x^2,

$$D(R) \geq \sigma_x^2\, 2^{-2R}. \qquad (3)$$

The work in [2] is focused on adapting codebook-based analysis-by-synthesis coding to account for the effect described by (2). Codebook contents are traditionally filtered to have the PSD described by $P_{xx}(\omega)$. [2] gives a procedure for filtering codebook contents to have the PSD described by $P_{\hat{x}\hat{x}}(\omega)$ since this may provide a more natural fit to actual coder operation. The process of moving from $P_{xx}(\omega)$ to $P_{\hat{x}\hat{x}}(\omega)$ is called reverse water-filling (RWF.) Due to complexities associated with deriving RWF filters, the work in [2] and here is limited to first-order auto-regressive (AR) signal modeling. Even this relatively simple approach improves the SNR of coded signals.

2. CODING CONSTRAINTS

Coding that minimizes a squared-error distortion measure is also called minimum mean-squared error (MMSE) coding. In MMSE coding, an input signal vector $\boldsymbol{x}$ is replaced by an approximation $\hat{\boldsymbol{x}}$ that minimizes the mean-squared error (MSE) between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ under some constraints on $\hat{\boldsymbol{x}}$. The nature of the constraints on $\hat{\boldsymbol{x}}$ determine the coding rate and the amount of coding distortion.

If $\hat{\boldsymbol{x}}$ is constrained to come from a linear subspace, then the MMSE coder projects $\boldsymbol{x}$ onto that subspace to find $\hat{\boldsymbol{x}}$. This projection minimizes the MSE between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, and insures that the error vector $\boldsymbol{e}$ is orthogonal to (and uncorrelated with) $\hat{\boldsymbol{x}}$.

In MMSE analysis-by-synthesis speech coding, $\hat{\boldsymbol{x}}$ is commonly constrained to come from a codebook or to be built from codebook members under some constraints. In general these constraints are quite different from the linear subspace constraint mentioned above. Codebook constraints often do not insure the orthogonality of $\boldsymbol{e}$ and $\hat{\boldsymbol{x}}$. Because of this (2) may be violated. In MMSE coding with codebook constraints, the relationship between $P_{\hat{x}\hat{x}}(\omega)$ and $P_{xx}(\omega)$ is driven by the distributions of input signals and codewords in the codebook. Two examples follow.

Uniform Density Codebook

We formed an *n*-dimensional codebook of *m* random codewords uniformly distributed throughout the hypersphere of radius 2. We then used this codebook to do exhaustive-search MMSE coding of zero-mean, unit-variance Gaussian vectors. For rates between 0.2 and 2.0 bits/sample ($R = \log_2(m)/n$) we found $P_{\hat{x}\hat{x}}(\omega) > P_{xx}(\omega)$ for all ω. This result is not consistent with (2). For insight, consider a small hyperspherical neighborhood *N* about the input vector $\boldsymbol{x}$ and divide this neighborhood into two regions. Let N^+ be the set of points in *N* for which the distance to the origin is greater than $|\boldsymbol{x}|$ and let N^- be the remainder of *N*. N^+ has greater hypervolume than N^- and the spatial density of codewords is uniform, so N^+ contains more codewords than N^-. Thus it is more likely that the codeword $\hat{\boldsymbol{x}}$ that is closest to $\boldsymbol{x}$ will be in N^+ rather than in N^-. On average we find that $|\hat{\boldsymbol{x}}|^2 > |\boldsymbol{x}|^2$ and hence $P_{\hat{x}\hat{x}}(\omega) > P_{xx}(\omega)$.

Gaussian Codebook

We repeated this experiment using a Gaussian codebook, and found $P_{\hat{x}\hat{x}}(\omega) < P_{xx}(\omega)$ for all ω, which can be consistent with (2). The Gaussian codebook is more densely populated near the origin and N^- typically contains more codewords than N^+. Thus, on average we find that $|\hat{\boldsymbol{x}}|^2 < |\boldsymbol{x}|^2$ and $P_{\hat{x}\hat{x}}(\omega) < P_{xx}(\omega)$. Since codebook-based speech coders often use codebooks that are Gaussian, or at least exhibit increasing codeword density towards the origin, it is likely that we will often find $P_{\hat{x}\hat{x}}(\omega) < P_{xx}(\omega)$ in speech coding situations.

It appears there may be two complications associated with RWF in codebook-based coding. First, ***e*** is not always orthogonal to $\hat{\boldsymbol{x}}$ so (2) is not always satisfied. Second, (3) gives a bound on *D* for any given rate *R* in the white Gaussian case, but it does not give actual values of *D* to use in RWF for general cases. Thus we elected to do some coding experiments to determine optimum levels of RWF for noise and speech signals. We investigated RWF using two measures of coding distortion, and we compared RWF with the application of a gain factor.

3. CODING EXPERIMENTS

[2] gives a procedure for filtering codebook contents to have the PSD described by $P_{\hat{x}\hat{x}}(\omega)$ as given in (2), when $P_{xx}(\omega)$ is a first-order AR process. For a given rate *R*, (3) is used (with equality) to determine a level of coding distortion *D*. This distortion level is then used to design a first-order auto-regressive moving average (ARMA) filter that will shape white codewords to the PSD described by $P_{\hat{x}\hat{x}}(\omega)$.

We adopted this filter design procedure, but rather than driving it with the actual coder rate *R*, we used an effective rate $\tilde{R}$. This allowed us to vary $\tilde{R}$ to determine an optimal level of RWF. To prevent attempts at designing filters for $P_{\hat{x}\hat{x}}(\omega) < 0$, we followed the procedure of [2] and calculated effective distortion $\tilde{D}$ as

$$\tilde{D}(\tilde{R}) = \min\left(\sigma^2\, 2^{-2\tilde{R}},\ \min_{\omega} P_{xx}(\omega)\right), \tag{4}$$

where *σ* is defined for each experimental case below.

For white signals RWF is equivalent to the application of a gain factor. If codeword variance is equal to input signal variance, then (2) and (3) give this gain factor as

$$G(\tilde{R}) = \sqrt{1 - 2^{-2\tilde{R}}} \quad . \tag{5}$$

Following [2], our experiments were done at *R*=0.25 and *R*=1 bits/sample. These rates nicely bracket the rates of much current speech coding work. For *R*=0.25 we used signal vectors and codewords with length *n*=40, and codebooks with $m = 2^{10}$ codewords. For *R*=1, we used *n*=10 and $m = 2^{10}$. Note that *R*=log$_2$(*m*)/*n*.

We used SNR and log-spectral error (LSE) as measures of coding distortion:

$$SNR = 10\log_{10}\left(\frac{|\boldsymbol{x}|^2}{|\boldsymbol{e}|^2}\right), \qquad LSE = \frac{1}{n}\sum_{i=1}^{n}\left|10\log_{10}\left(\frac{P_{\hat{x}\hat{x}}(\omega_i)}{P_{xx}(\omega_i)}\right)\right| \ , \tag{6}$$

where $P_{\hat{x}\hat{x}}(\omega_i)$ and $P_{xx}(\omega_i)$ represent length *n* sampled PSD's calculated using a symmetric Hamming window. For the white and colored noise coding experiments, SNR and LSE were calculated for length *n* signal vectors. For the speech coding experiments, SNR and LSE were calculated over 10 ms frames of the speech signals, consistent with [2]. Experimental results are given in Table 1. Searches for optimal values G_{opt} and $\tilde{R}_{opt}$ used step size 0.01.

White Gaussian Noise

The first set of coding experiments used zero-mean, unit-variance white Gaussian input vectors and codewords. These codewords were scaled by the gain factor $G(\tilde{R})$ which is equivalent to RWF at an effective rate $\tilde{R}$ when $G \le 1$. Given an input signal vector ***x***, the coder exhaustively searched the scaled codebook to find the codeword $\hat{\boldsymbol{x}}$ that minimized $|\boldsymbol{e}|^2 = |\boldsymbol{x} - \hat{\boldsymbol{x}}|^2$. We varied *G* over a range to determine what values of *G* would give maximal SNR and minimal LSE. SNR and LSE values were averaged over 20,000 trials for each value of *G*. The table shows that in the white Gaussian case, gain or RWF can improve SNR by about 1.0 dB at $R = 0.25,\ \tilde{R} = 0.18$ and by 0.2 dB at $R = 1,\ \tilde{R} = 0.68$. On the other hand, this technique causes LSE to increase by 2.3 dB and 0.2 dB respectively. The table also shows the results of optimizing RWF to minimize LSE. This can only be done at the expense of SNR. An SNR vs. LSE trade-off must be made. In several cases LSE is minimized by gains greater than 1. This corresponds to "water filling" rather than RWF. This result indicates that in some cases the best spectral match is accomplished by fighting, rather than accommodating the effect described in (2).

Colored Gaussian Noise

The second set of coding experiments used zero-mean, colored Gaussian input vectors. Unit-variance white Gaussian vectors were colored using a low-pass first-order AR filter

$$H(z) = \frac{\sigma G}{1 - \alpha_1 z^{-1}} \quad , \tag{7}$$

with *σ* = *G* = 1. We used α_1=0.7 for consistency with [2] and with speech PSD's. Codewords were constructed by filtering unit-variance white Gaussian vectors as well. For the RWF case we used the ARMA filter specified in [2]:

$$H(z) = \frac{\beta_0 - \beta_1 z^{-1}}{1 - \alpha_1 z^{-1}}, \qquad \beta_0 = \sqrt{\frac{1}{2}\left(\lambda + \sqrt{\lambda^2 - 4\tilde{D}(\tilde{R})^2\alpha_1^2}\right)}, \tag{8}$$
$$\beta_1 = -\frac{\tilde{D}(\tilde{R})\alpha_1}{\beta_0}, \qquad \lambda = \sigma^2 - \tilde{D}(\tilde{R}) - \tilde{D}(\tilde{R})\alpha_1^2 \ ,$$

with *σ* = 1. The MA portion of this filter accomplishes RWF.

Given the simplicity of the gain factor approach and its equivalence to RWF in the white case, we applied it in this case as well. We created a second set of filtered codewords using (7) with *σ* = 1. We varied *G* and $\tilde{R}$ over ranges to determine what values would optimize SNR and LSE. SNR and LSE values were averaged over 5000 or 100,000 trials.

With $\tilde{R} = R$, we were able to approximately reproduce the SNR improvements reported in [2]. The absolute SNR values were not in exact agreement but the improvements were 0.7 dB at $R = \tilde{R} = 0.25$ and 0.4 dB at $R = \tilde{R} = 1$ which agree well with the 0.7 dB and 0.3 dB improvements reported in [2]. The best-case SNR increases were 0.7 dB at $\mathrm{R} = 0.25,\ \tilde{R} = 0.77$ and 0.5 dB at $R = 1,\ \tilde{R} = 0.85$. The peaks in the SNR vs. $\tilde{R}$ curves are broad, so while *R* and $\tilde{R}_{opt}$ are not particularly close, the resulting SNR's are quite close. The optimization of $\tilde{R}$ offers little advantage over the case $\tilde{R}=R$. At R=0.25, the gain factor and RWF offer similar SNR improvements but at R=1 RWF offers a larger SNR improvement than the gain factor. At both rates, the gain factor gives a better LSE than RWF does. An SNR vs. LSE trade-off must be made.

Speech

The final set of coding experiments used 64 seconds of high-quality speech signals with 300-3400 Hz nominal passband and f_s=8 kHz. Two female and two male English-language speakers were used and each speaker provided six sentences. The autocorrelation method (using a 20 ms Hamming window) was used to derive a first-order AR model for each 10 ms frame of speech. Each frame was then divided into 2 (R=0.25) or 8 (R=1) signal vectors. Codewords were constructed by filtering unit-variance white Gaussian vectors. For the RWF case we used the filter specified in (8) and for the gain factor case we used the filter specified in (7). In each case σ and α_1 were set to agree with the first-order AR speech model. We varied $\tilde{R}$ and G over ranges to determine what values would optimize SNR and LSE. For each value of $\tilde{R}$ or G the SNR and LSE values were averaged over the frames of all 24 sentences, excluding any frames with energy 20 dB or more below the average frame energy, consistent with [2].

The table shows that for these speech signals, RWF can provide best-case SNR improvements of 0.5 dB and 0.3 dB at R=0.25 and R=1 respectively. The improvements reported in [2] are 0.5 and 0.4 dB so our optimization procedure offers no benefit in this case. RWF also improves LSE by about 1.5 and 0.6 dB respectively. At R=0.25, the gain factor gives the same SNR improvement as RWF, but a smaller LSE improvement. At R=1 the gain factor provides no benefit. With these speech signals, the maximal SNR and minimal LSE operating points are quite close, so there is little need or opportunity to trade one off against the other.

4. CONCLUSIONS

Codebook constrained MMSE coding may or may not motivate RWF, depending on the signal and codeword distributions. For speech coding, these distributions do generally motivate RWF. For white and low-passed Gaussian signals, RWF can increase SNR but only at the expense of LSE. For white signals RWF is equivalent to a gain factor. For low-passed Gaussian signals a gain factor may offer some advantages over RWF in terms of LSE. For speech signals, RWF is more useful than a gain factor. RWF based on an optimized effective rate offers no significant advantage over the use of actual rates. We offer thanks to Søren Vang Andersen and W. Bastiaan Kleijn, authors of [2]. This paper relies heavily on their work.

REFERENCES

[1] T. Berger, *Rate Distortion Theory – A Mathematical Basis for Data Compression*, Englewood Cliffs, NJ, Prentice-Hall, 1971.

[2] S.V. Andersen & W.B. Kleijn, "Reverse Water-Filling in Predictive Encoding of Speech," *Proc. 1999 IEEE Workshop on Speech Coding*, Porvoo, Finland, June 1999, pp. 105-107.

Signal, Rate	Baseline SNR (dB)	Baseline LSE (dB)	Gain			Reverse Water-Filling		
			G_{opt}	SNR (dB)	LSE (dB)	$\tilde{R}_{opt}$	SNR (dB)	LSE (dB)
White Gaussian, R=0.25	0.12	5.81	G_{maxSNR}=0.47	1.16	8.12	$\tilde{R}_{maxSNR}$ =0.18	1.16	8.12
			G_{minLSE}=1.17	-0.48	5.75	$\tilde{R}_{minLSE} \rightarrow \infty$	0.12	5.81
Colored Gaussian, R=0.25	1.71	5.83	G_{maxSNR}=0.60	2.46	6.94	$\tilde{R}_{maxSNR}$ =0.77	2.43	7.87
			G_{minLSE}=1.16	1.15	5.69	$\tilde{R}_{minLSE} \rightarrow \infty$	1.71	5.83
Speech, R=0.25	2.17	11.21	G_{maxSNR}=0.63	2.70	10.29	$\tilde{R}_{maxSNR}$ =0.48	2.69	9.73
			G_{minLSE}=0.46	2.54	10.10	$\tilde{R}_{minLSE}$ =0.74	2.68	9.65
White Gaussian, R=1	4.54	4.79	G_{maxSNR}=0.78	4.77	5.00	$\tilde{R}_{maxSNR}$ =0.68	4.77	5.00
			G_{minLSE}=0.98	4.59	4.75	$\tilde{R}_{minLSE}$ =2.33	4.59	4.75
Colored Gaussian, R=1	6.35	4.84	G_{maxSNR}=0.80	6.60	4.97	$\tilde{R}_{maxSNR}$ =0.85	6.80	5.70
			G_{minLSE}=1.06	6.22	4.68	$\tilde{R}_{minLSE} \rightarrow \infty$	6.35	4.84
Speech, R=1	7.05	9.18	G_{maxSNR}=0.93	7.06	9.10	$\tilde{R}_{maxSNR}$ =1.04	7.35	8.62
			G_{minLSE}=0.68	6.71	8.93	$\tilde{R}_{minLSE}$ =0.90	7.24	8.54

Table 1. Results of coding experiments with gain or reverse water-filling.