Ch. 13 Transform Coding

Example, Insight & Algorithms

Example of Bit Allocation *Theory*

- Consider Scalar Quantizers
- Need function for $D_i(R_i)$
  - Use “High-Rate” Approximation

This is similar to what we derived for USQ of Uniform RV

$$D_i(R_i) = C_i \sigma_i^2 2^{-2R_i}$$

$C_i =$ pdf-dependent constant

$\sigma_i^2 =$ variance of i^{th} transform coefficient

$R_i =$ # o fbits allocated to i^{th} quantizer

- Assume we know the variances σ_i^2 of the transform coefficients
  - In Theory: assume WSS & choose PDF model & ACF model, then use analysis to get σ_i^2 for the chosen transform
  - In Practice: One way is to collect a large set of typical signals and estimate variances by averaging over the set of coefficients (“pseudo ensemble”)… Another way… TC is often applied on a block-by-block basis so can average coefficients over several blocks

Assume all coefficients have the same type of PDF, just different variances

$$\Rightarrow \quad C_i = C \quad \forall i \qquad \Rightarrow \quad D_i(R_i) = C\sigma_i^2 2^{-2R_i}$$

Now… "Equal Slopes…" says to set $\underbrace{\frac{\partial D_i(R_i)}{\partial R_i}}_{\text{So find this}} = -\lambda$

➡ $$\frac{\partial D_i(R_i)}{\partial R_i} = \underbrace{C\sigma_i^2(\ln 2)(-2)2^{-2R_i} = -\lambda}_{\text{Solve for } R_i}$$

➡ $$\boxed{R_i = \underbrace{\tfrac{1}{2}\log_2\left[2C\sigma_i^2(\ln 2)\right]}_{\text{Provides the variation w/ } i} \underbrace{- \tfrac{1}{2}\log_2 \lambda}_{\text{Additive Constant}}} \qquad (\star)$$

➔ Shifts up/down

➔ Adjust λ until $\Sigma R_i = R_B$

➡ Now… add up R_is and set = to R_B… then solve for λ

So…

$$R_B = \sum_{i=0}^{N-1} R_i = \underbrace{\tfrac{1}{2}\sum_{i=0}^{N-1} \log_2\left[2C\sigma_i^2(\ln 2)\right]}_{= \tfrac{1}{2}\log_2\left[2C(\ln 2)\prod_{i=0}^{N-1}\sigma_i^2\right]} - \tfrac{N}{2}\log_2 \lambda$$

The negative of this is the slope that all the Qs should have for an optimal allocation of R_B bits

Solve for λ:

$$\boxed{\lambda = \frac{2C(\ln 2)\left(\prod_{i=0}^{N-1}\sigma_i^2\right)^{1/N}}{2^{2R_B/N}}} \qquad (\bigstar\bigstar)$$

Putting (★★) into (★) gives the optimal allocation (see next page for steps):

$$R_i = \tfrac{1}{2}\log_2\left[2C\sigma_i^2(\ln 2)\right] - \tfrac{1}{2}\log_2 \lambda \qquad (\bigstar)$$

Indicates Optimal

$$R_k^* = \tfrac{1}{2}\log_2\left[2C\sigma_k^2(\ln 2)\right] - \tfrac{1}{2}\log_2\left[\frac{2C(\ln 2)\left(\prod_{i=0}^{N-1}\sigma_i^2\right)^{1/N}}{2^{2R_B/N}}\right]$$

Now do various manipulations using properties of logarithms…

$$R_k^* = \tfrac{1}{2}\log_2\left[\sigma_k^2\right] + \tfrac{1}{2}\log_2\left[2\cancel{C}(\ln 2)\right] - \tfrac{1}{2}\log_2\left[2\cancel{C}(\ln 2)\right] - \tfrac{1}{2}\log_2\left[\frac{\left(\prod_{i=0}^{N-1}\sigma_i^2\right)^{1/N}}{2^{2R_B/N}}\right]$$

$$= \tfrac{1}{2}\log_2\left[\frac{\sigma_k^2\, 2^{2R_B/N}}{\left(\prod_{i=0}^{N-1}\sigma_i^2\right)^{1/N}}\right]$$

$$= \tfrac{1}{2}\log_2\left[2^{2R_B/N}\right] + \tfrac{1}{2}\log_2\left[\frac{\sigma_k^2}{\left(\prod_{i=0}^{N-1}\sigma_i^2\right)^{1/N}}\right]$$

$$R_k^* = \frac{R_B}{N} + \frac{1}{2}\log_2\left[\frac{\sigma_k^2}{\left(\prod_{i=0}^{N-1}\sigma_i^2\right)^{1/N}}\right] \qquad (\star\star\star)$$

$\frac{R_B}{N} \triangleq \bar{R}$

This term gives the “evenly divided” # of bits

This term tells how R_k should differ from the “evenly divided” value

Denom = “geometric mean” of σ_i^2’s $\triangleq \gamma$

$$\gamma = \left(\prod_{i=0}^{N-1}\sigma_i^2\right)^{1/N}$$

- Coefficients w/ $\sigma_k^2 >$ geometric mean… get more bits than $\bar{\text{R}}$ bits
- Coefficients w/ $\sigma_k^2 <$ geometric mean… get fewer bits than $\bar{\text{R}}$ bits

Note: Allocation does not depend on the pdf-type-dependent constant C

➔ **Depends only on the PDF variances**

$$D_i(R_i) = C\sigma_i^2 2^{-2R_i}$$

Q: When we use this optimal allocation what is the resulting distortion?

A: Plug optimal allocation into assumed distortion function (here we've used the "high rate" approximation for a scalar quantizer):

$$D_k(R_k^*) = C\sigma_k^2 2^{-2R_k^*} = C\sigma_k^2 2^{-2\left(\bar{R} + \frac{1}{2}\log_2\left[\sigma_k^2/\gamma\right]\right)}$$

$$= C\sigma_k^2 2^{-2\left(\bar{R} + \frac{1}{2}\log_2\left[\sigma_k^2/\gamma\right]\right)} = C\sigma_k^2 2^{-2\bar{R}} 2^{\log_2\left[\gamma/\sigma_k^2\right]}$$

$$= C\sigma_k^2 \frac{\gamma}{\sigma_k^2} 2^{-2\bar{R}} = C\gamma 2^{-2\bar{R}}$$

Interesting! All Qs have ***same*** distortion!!

$$\boxed{D_k(R_k^*) = C\gamma 2^{-2\bar{R}}} \quad (\bigstar\bigstar\bigstar\bigstar)$$

Total Distortion is: $$\boxed{D = \sum_{i=0}^{N-1} D_i = NC\gamma 2^{-2\bar{R}}}$$

Distortion depends on: N, C, $\bar{R}$, γ

Not really important here

"Fixed"

Controlled by Choice of transform!! (See Next Slide)

Insight into ON Transform Choice

Total Distortion is: $D = NC\gamma 2^{-2\bar{R}}$ $\gamma = \left(\prod_{i=0}^{N-1} \sigma_i^2 \right)^{1/N}$

Controlled by Choice of Transform

Want to minimize γ!!

“Theorem” **Must choose ON transform to give “widely varying” σ_i^2 values**

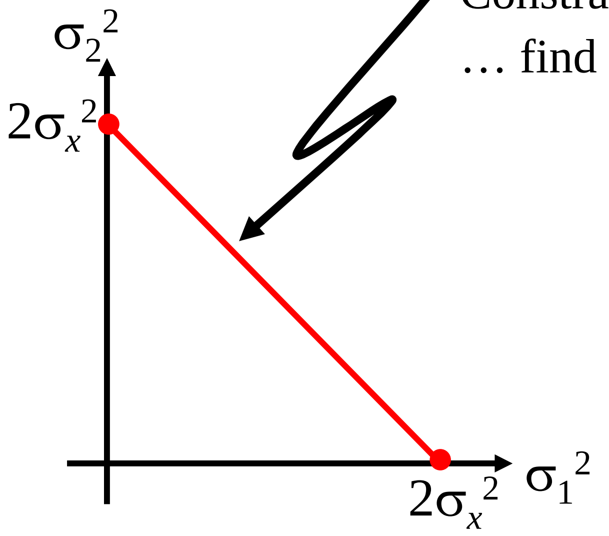
“*Proof*” Consider the N = 2 case. Let σ_x^2 be the variance of the original signal.

Since the transform is ON $\sigma_x^2 = \frac{1}{2}\left(\sigma_1^2 + \sigma_2^2\right)$ variances of transform coeff’s

(We’ll see this result later)

Constrained to this line segment…

… find where $\gamma = (\sigma_1^2 \sigma_2^2)^{1/2}$ is minimized



γ is Minimum (= zero) at end points:

- $\sigma_1^2 = 0$ & $\sigma_2^2 = 2\sigma_x^2$
- $\sigma_1^2 = 2\sigma_x^2$ & $\sigma_2^2 = 0$

γ is Maximum (= σ_x^2) in center:

- $\sigma_1^2 = \sigma_x^2$ & $\sigma_2^2 = \sigma_x^2$

One Last Pair of Insights

Raising 2 to each side of (★★★) we get:

$$(\bigstar\bigstar\bigstar)\quad \boxed{R_k^* = \bar{R} + \tfrac{1}{2}\log_2\left[\frac{\sigma_k^2}{\gamma}\right]} \Rightarrow \boxed{2^{R_k^*} = \left(\frac{2^{\bar{R}}}{\sqrt{\gamma}}\right)\sigma_k}$$

Insight #1: (# of Quantization Levels of k^{th} quantizer) ~ k^{th} Std. Dev

Insight #2:

If… $\sigma_i = 2\,\sigma_j$

Then… $R_i = R_j + 1$

Bit Allocation Algorithms

Method #1: Based on Theoretical Allocation (★★★)

$$\boxed{R_k^* = \bar{R} + \tfrac{1}{2}\log_2\left[\frac{\sigma_k^2}{\gamma}\right]}$$

Get Est of Vars of Coeffs

1. Collect "training" set of L signals that typify the class of signals of interest

   $x_l[n] \quad l = 1, 2, 3, \dots, L \qquad n = 0, 1, 2, \dots, N\text{-}1$

2. Transform "training" signals into "training" coefficients

   $\mathbf{y}_l = \boldsymbol{A}\,\mathbf{x}_l$

3. Estimate the variance of each coefficient by averaging over training set

   $$\hat{\sigma}_k^2 = \frac{1}{L}\sum_{l=1}^{L}\left(y_l[k]\right)^2$$ (Assumes zero mean)

4. Compute optimal allocation values R_k^* using (★★★)

   Uh Oh!!! The resulting R_k^* values can be:

   - Non-Integer Valued
   - Negative Valued

5. Round to integers
6. Set negative allocations to zero….total allocation now exceeds R_B

   ➔ "Un-allocate" some bits to get back down to R_B

Method #2: Based on “*One Last Pair of Insights*”

If… $\sigma_i = 2\,\sigma_j$

Then… $R_i = R_j + 1$

Also based on that D_i is the same for all quantizers…See (★★★★).

For the “high-rate approximation” we have

$$\sqrt{D_i} = \sqrt{C}\,\frac{\sigma_i}{2^{R_i^*}} \quad \Rightarrow \quad \frac{\sigma_i}{2^{R_i^*}} \quad \text{should be same for all } i$$

1. Estimate Variances of Training Set (See Steps 1 – 3 of Method #1)

   Get Estimated Std. Devs from Estimated variances: $\hat{\sigma}_k = \sqrt{\hat{\sigma}_k^2}$

2. Allocate a bit to the quantizer w/ largest $\hat{\sigma}_k$

   Then set that $\hat{\sigma}_k \leftarrow \hat{\sigma}_k / 2$

Book Error on p. 408… it divides variance by 2 rather than Std Dev

3. Stop if all bits are allocated…. Otherwise: Go to Step #2

This is a so-called “Greedy” algorithm…

At each step, a “greedy” algorithm takes the step that gives maximum improvement… but there is no guarantee it will find the optimum.

Methods #1 & #2 use theoretical results found via the “*high-rate approximation*” for the distortion of *scalar quantizers*

In many scenarios these two assumptions may not be valid!

Method #3: A Generalization of Method #2

1. Estimate Variances of Training Set (See Steps 1 – 3 of Method #1)
   Get Estimated Std. Devs from Estimated variances:
2. Also using the Training Set… “Develop” functions for the distortions
   $D_i(R_i) = f_i(R_i,\sigma_i^2)$
3. Set all $R_i = 0$ & Calculate $D_i(0)$ for all i
4. Allocate a bit to the quantizer with the largest D_i
   Increment that R_i
   Re-compute that D_i
5. Stop if all bits are allocated… Otherwise, Go to Step #4…

Methods #1 - #3 are all based on ***Classical*****,** ***Average*** ***R-D Theory***

i.e., variances are estimated over a "pseudo-ensemble" of a training set

➔ Bit allocations are done once based on training set variances
➔ Get optimal distortion "on average"
➔ Some signals have distortions far above the average distortion

Operational R-D Methods try to give the best distortion possible for the *particular signal* being coded

Method #4: Operational R-D Version of Method #3

Instead of a *function* for $D_i(R_i)$, we *measure distortion* for a *specific allocated* R_i using the *specific signal*

1. Set all $R_i = 0$ (or to the minimum allowed R_i)
2. Measure $D_i(R_i)$ for all i
3. Allocate a bit to the quantizer with the largest D_i
   Re-Measure D_i with the newly allocated bit.
4. Stop if all bits are allocated… Otherwise, Go to Step #3…

For an even better algorithm… See "Shoham & Gersho Paper" mentioned earlier