Discrete-Cosine Transform

The *M*-pt DCT pair is given by

$$U(m) = \kappa_\mu \sum_{\nu=0}^{M-1} \upsilon(\nu)\chi o\sigma\left[\frac{(2\nu+1)\mu\pi}{2M}\right],\ 0 \le \mu \le M-1$$

$$\upsilon(\nu) = \frac{2}{M}\sum_{\mu=0}^{M-1} \kappa_\mu Y(\mu)\chi o\sigma\left[\frac{(2\nu+1)\mu\pi}{2M}\right],\ 0 \le \nu \le M-1$$

where

$$k_m = \begin{cases} 1/\sqrt{2}, & \mu = 0 \\ 1, & 1 \le \mu \le M-1 \end{cases}.$$

Figure 10.3.

Sliding DCT

The DCT that we compute on the input vector, uses a sliding window so that the computation is performed for each new incoming sample as opposed to a block-style computation.

We wish to develop an efficient algorithm for computing the sliding DCT. By efficient we want to avoid having to recompute the DCT of **u**(*n*) but rather take the DCT of **u**(*n* – 1) and add, subtract the effects of *u*(*n*), *u*(*n* – *M*), i.e. recursively update.

We'll proceed with the following steps

1) Show how the DFT is related to the DCT.
2) Exploit the above relation to recursively update the DCT of the current signal vector.

As a reminder, the *M*-pt Discrete Fourier Transform (DFT) pair is defined as

$$X(k) = \sum_{\nu=0}^{M-1} \xi(\nu)\Omega_N^{\kappa\nu},\ 0 \le \kappa \le M-1$$

$$\xi(\nu) = \frac{1}{M}\sum_{\kappa=0}^{M-1} \Xi(\kappa)\Omega_N^{-\kappa\nu},\ 0 \le \nu \le M-1$$

where

$$W_N = \varepsilon^{-\phi 2\pi/N}.$$

Step 1: Consider the input vector

$$\mathbf{u}(n) = [\,\upsilon(\nu) \quad \Lambda \quad \upsilon(\nu - M + 1)]^T$$

and its even extension **a**(*i*) [i.e. reflect **u**(*n*) about *u*(*n* – *M* + 1)] where

$$a(i) = \begin{cases} \upsilon(\iota), & \nu \le \iota \le \nu - M + 1 \\ \upsilon(-\iota + 2\nu - 2M + 1), & \nu - M \le \iota \le \nu - 2M + 1 \end{cases}.$$

The *m*th element of the 2*M*-point DFT of **a**(*i*) is defined by

$$A_m(n) = \sum_{i=n-2M+1}^{n} a(i) W_{2M}^{m(n-i)} \; .$$

We now segment the above summation into two summations

$$\begin{aligned} A_m(n) &= \sum_{i=n-M+1}^{n} a(i) W_{2M}^{m(n-i)} + \sum_{i=n-2M+1}^{n-M} a(i) W_{2M}^{m(n-i)} \\ &= \sum_{i=n-M+1}^{n} u(i) W_{2M}^{m(n-i)} + \sum_{i=n-2M+1}^{n-M} u(-i+2n-2M+1) W_{2M}^{m(n-i)} \\ &= \sum_{i=n-M+1}^{n} u(i) W_{2M}^{m(n-i)} + \sum_{i=n-M+1}^{n} u(i) W_{2M}^{m(i-n+2M-1)} \; ; -i \leftarrow i-2n+2M-1 \\ &= A_m^{(1)}(n) + A_m^{(2)}(n) \end{aligned} \; .$$

We then have

$$\begin{aligned} A_m(n) &= W_{2M}^{m(M-1/2)} \sum_{i=n-M+1}^{n} u(i) \left[W_{2M}^{-m(i-n+M-1/2)} + W_{2M}^{m(i-n+M-1/2)} \right] \\ &= 2(-1)^m \, W_{2M}^{-m/2} \sum_{i=n-M+1}^{n} u(i) \cos\left[\frac{m(i-n+M-1/2)}{M} \pi \right] \end{aligned}$$

which for $0 \le m \le M-1$ (and except for the scaling factor in front) is the *M*-point DCT of **u**(*n*)

$$C_m(n) = k_m \sum_{i=n-M+1}^{n} u(i) \cos\left[\frac{m(i-n+M-1/2)}{M} \pi \right] \; .$$

Step 2: Now consider the recursive update of $A_m(n) = A_m^{(1)}(n) + A_m^{(2)}(n)$.

$$\begin{aligned} A_m^{(1)}(n) &= \sum_{i=n-M+1}^{n} u(i) W_{2M}^{m(n-i)} \\ &= u(n) + \sum_{i=n-M+1}^{n-1} u(i) W_{2M}^{m(n-i)} \\ &= u(n) + \underbrace{\sum_{i=n-M}^{n-1} u(i) W_{2M}^{m(n-i)}}_{A_m^{(1)}(n-1)} - u(n-M) W_{2M}^{m(M)} \\ &= A_m^{(1)}(n-1) + u(n) - (-1)^m \, u(n-M) \end{aligned} \; .$$

In a similar way we can show

$$A_m^{(2)}(n) = W_{2M}^{-m} A_m^{(2)}(n-1) + W_{2M}^{-m} \left[u(n) - (-1)^m \, u(n-M) \right] .$$

Eigenvalue Estimation

In order to complete the DCT-LMS we must estimate Λ^{-1}. We first define the sample correlation matrix (or average correlation matrix) as (assuming ergodicity)

$$\hat{\mathbf{R}}(n) = \frac{1}{\nu}\sum_{\iota=1}^{\nu}\boldsymbol{\upsilon}(\iota)\boldsymbol{\upsilon}^{T}(\iota)\,.$$

The coefficients of the DCT matrix $\hat{\mathbf{Q}}$, provide an approximation to $\mathbf{Q}^T$. Thus we have

$$\begin{aligned}\hat{\mathbf{v}}(n) &= \Delta\mathrm{XT}[\boldsymbol{\upsilon}(\nu)] \\ &= [X_0(\nu) \quad \Lambda \quad X_{M-1}(\nu)]^T \\ &= \hat{\boldsymbol{\Theta}}\boldsymbol{\upsilon}(\nu)\end{aligned}\,.$$

Now we use these approximations to write

$$\begin{aligned}\hat{\Lambda}(\nu) &= \hat{\boldsymbol{\Theta}}\hat{\mathbf{P}}\hat{\boldsymbol{\Theta}}^T \\ &= \frac{1}{\nu}\sum_{\iota=1}^{\nu}\hat{\boldsymbol{\Theta}}\boldsymbol{\upsilon}(\iota)\boldsymbol{\upsilon}^T(\iota)\hat{\boldsymbol{\Theta}}^T \\ &= \frac{1}{\nu}\sum_{\iota=1}^{\nu}\hat{\boldsymbol{\varpi}}(\iota)\hat{\boldsymbol{\varpi}}^T(\iota)\end{aligned}\,.$$

It is unlikely that $\hat{\Lambda}(\nu)$ is strictly diagonal but we'll use the approximation. We then have eigenvalue estimates at time *n*,

$$\hat{\lambda}_m(n-1) = \frac{1}{n-1}\sum_{i=1}^{n-1}C_m^2(i),\ \ 0 \le m \le M-1\,.$$

Clearly, the eigenvalue estimates can be recursively updated with

$$\hat{\lambda}_\mu(\nu) = \hat{\lambda}_\mu(\nu-1) + \frac{1}{\nu}\left[X_\mu^2(\nu) - \hat{\lambda}_\mu(\nu-1)\right]\,.$$

Finally, we can modify the above for operation in a non-stationary environment by applying a forgetting factor $0 < \gamma < 1$

$$\hat{\lambda}_\mu(\nu) = \gamma\hat{\lambda}_\mu(\nu-1) + \frac{1}{\nu}\left[X_\mu^2(\nu) - \gamma\hat{\lambda}_\mu(\nu-1)\right]\,.$$

Table 10.2: DCT-LMS