Blind KLT Coding

Carlos E. Davila
Electrical Engineering Department, Southern Methodist University
Dallas, Texas 75275-0338

Abstract

We describe several methods for coding signals via the Karhunen-Loeve Transform (KLT) which do not require the encoding and transmission of the KLT basis vectors.

1 Introduction

The Karhunen-Loeve Transform (KLT) is known to be the optimum transform for signal compression [1]. Unfortunately, the KLT basis functions, which are the eigenvectors of the data autocorrelation matrix, are data dependent. Hence the basis functions must also be encoded and transmitted which reduces compression and leads to increased data rates. For this reason, the KLT has found limited use in data compression applications. Let

$$x_n = \left[\begin{array}{cccc} x(nN-1) & x(nN-2) & \cdots & x(N(n-1)) \end{array} \right]^T \tag{1}$$

be the N-dimensional signal frame to be encoded. We assume that x_n has autocorrelation matrix $R = E\left[x_n x_n^T\right]$ having rank $r \leq N$. This means that x_n can be represented as a linear combination of the eigenvectors of R given by $q^1, q^2, \ldots, q^r$, corresponding to eigenvalues $\lambda^1 \geq \lambda^2 \geq \ldots \geq \lambda^r > 0$, respectively. Let $Q = \left[\begin{array}{cccc} q^1 & q^2 & \cdots & q^r \end{array} \right]$ be an $N \times r$ matrix whose columns are the KLT basis vectors (eigenvectors of R). The transform coefficients, given by $y_n = Q^T x_n$, can then be quantized as $\hat{y}_n$, encoded, and transmitted. If the receiver has knowledge of the basis vectors Q, x_n can be recovered as $\hat{x}_n = Q\hat{y}_n$. If the signal $x(n)$ is statistically stationary then the eigenvectors need only be estimated and transmitted once, which would not lead to much loss of compression, however in practice, the eigenstructure of most signals tends to vary considerably over time. Hence the eigenvectors of R need to be constantly retransmitted which is why the KLT is not often used. In this paper, we give a method of determining the basis vectors for the KLT directly from the KLT coefficients given only very limited knowledge of $x(n)$. This eliminates the need to retransmit the KLT basis vectors.

2 Tracking KLT Basis Vectors

Blind estimation of the KLT basis vectors can be accomplished using ideas from the subspace tracking literature. Let $\hat{R}_n$ be an estimate of $E\left[x_n x_n^T\right]$ that is updated using,

$$\hat{R}_n = \gamma \hat{R}_{n-1} + x_n x_n^T \tag{2}$$

where $0 < \gamma < 1$. Let

$$\hat{Q}_n = \left[\begin{array}{cccc} \hat{q}_n^1 & \hat{q}_n^2 & \cdots & \hat{q}_n^r \end{array} \right] \tag{3}$$

and $\hat{\Lambda}_n = diag(\hat{\lambda}_n^1, \hat{\lambda}_n^2, \ldots, \hat{\lambda}_n^r)$ be *estimates* of the eigenvectors and eigenvalues, respectively, of $\hat{R}_n$. Then $\hat{R}_n \approx \gamma \hat{Q}_{n-1} \hat{\Lambda}_{n-1} \hat{Q}_{n-1}^T + x_n x_n^T$. The eigenvector estimates can be updated as follows:

sender	*receiver*
$\hat{Q}_0 = I_N(:, 1:r)$ *for* $n = 1, 2, \ldots$ $\overline{Q}_n = \left[\begin{array}{cc} \hat{Q}_{n-1} & v_n \end{array} \right]$ $y_n = \hat{Q}_{n-1}^T x_n$ $\overline{y}_n = \left[\begin{array}{cc} y_n^T & x_n^T v_n \end{array} \right]^T$ *transmit* $\overline{y}_n$ *to receiver* $F = \gamma \overline{Q}_n^T \hat{Q}_{n-1} \hat{\Lambda}_{n-1} \hat{Q}_{n-1}^T \overline{Q}_n + \overline{y}_n \overline{y}_n^T$ $G = \overline{Q}_n^T \overline{Q}_n$ *solve* $FW_n = GW_n \Pi_n$ $\hat{Q}_n = \overline{Q}_n W_n(1:r)$ *end*	$\hat{Q}_0 = I_N(:, 1:r)$ *for* $n = 1, 2, \ldots$ $\overline{Q}_n = \left[\begin{array}{cc} \hat{Q}_{n-1} & v_n \end{array} \right]$ *wait for* $\overline{y}_n$ $\hat{x}_n = \hat{Q}_{n-1} y_n$ $F = \gamma \overline{Q}_n^T \hat{Q}_{n-1} \hat{\Lambda}_{n-1} \hat{Q}_{n-1}^T \overline{Q}_n + \overline{y}_n \overline{y}_n^T$ $G = \overline{Q}_n^T \overline{Q}_n$ *solve* $FW_n = GW_n \Pi_n$ $\hat{Q}_n = \overline{Q}_n W_n(1:r)$ *end*

Table 1: Algorithm AKLT1 for blind estimation of KLT basis vectors. Both the sender and the receiver must run the algorithms in unison. The search direction vector v_n is assumed known to both the sender and receiver for each n.

1. Solve the generalized eigenvalue problem
$$FW_n = GW_n \Pi_n \tag{4}$$
where $F = \overline{Q}_n^T \left(\gamma \hat{Q}_{n-1} \hat{\Lambda}_{n-1} \hat{Q}_{n-1}^T + x_n x_n^T \right) \overline{Q}_n$, $G = \overline{Q}_n^T \overline{Q}_n$, and W_n and the diagonal matrix Π_n are the respective generalized eigenvectors and eigenvalues. The matrix $\overline{Q}_n = \left[\begin{array}{cc} \hat{Q}_{n-1} & v_n \end{array} \right]$ has dimension $N \times (r+1)$ and v_n is a *search direction vector*.

2. Update the eigenvector estimates as
$$\hat{Q}_n = \overline{Q}_n W_n(1:r) \tag{5}$$
where $W_n(1:r)$ are the eigenvectors corresponding to the maximum r eigenvalues in Π_n.

3. The eigenvalue estimates are updated as $\hat{\Lambda}_n = \Pi_n(1:r, 1:r)$.

If, in the above algorithm, the search direction is set to $v_n = x_n$, then the algorithm is a standard *subspace averaging* algorithm used for subspace tracking [2]. Note that if we treat the columns of $\hat{Q}_{n-1}$ as the KLT basis vectors, then the KLT coefficients are contained in the first r elements of $\overline{Q}_n^T x_n$, hence the algorithm never explicitly uses x_n. It can be shown that if v_n is a white noise vector, independent of x_n, then the eigenvectors of $\hat{R}_n$ can still be tracked. This implies that the above algorithm can be run by both the sender and the receiver concurrently using the same initial conditions. If the search direction vectors are known to both the sender and the receiver, then the receiver can also track the KLT basis vectors having only knowledge of the KLT coefficients and the additional scalar coefficient, $v_n^T x_n$. In this context, the algorithm is "blind" since the receiver requires no explicit knowledge of the signal $x(n)$ to track the KLT basis vectors. Table 1 lists the algorithm (AKLT1), I_N is the $N \times N$ identity matrix. The algorithm's convergence is proven in [3], where it is shown that faster convergence results when the estimated signal subspace dimension, r, is large. And when $r = N - 1$, the search space is the entire space which leads to an algorithm which exactly computes the eigenvectors and eigenvalues of the sample autocorrelation matrix.

3 Increasing Convergence Speed

Algorithm AKLT1 assumes r, the dimension of the signal subspace, is known. Some signals of practical interest have a signal subspace dimension which changes with time. Moreover, during sudden changes in the statistics of the signal, the accuracy of the existing signal subspace estimates can be poor, leading to

sender	receiver

$\hat{Q}_0 = I_N$
$for\ n = 1, 2, \ldots$
    $y_n = \hat{Q}_{n-1}^T x_n$
    $\hat{y}_n = \Delta(y_n)$
    $\rho = 1, k = 1$
    $while\ \rho\ >\ MSE_{max}$
        $\hat{x}_n = \hat{Q}_{n-1}(:, 1:k)\hat{y}_n(1:k);$
        $\rho = \|\hat{x}_n - x_n\|^2 / \|x_n\|^2$
        $k = k + 1;$
        $if\ k = N + 1\ AND\ \rho > MSE_{max}$
            $Orthonormalize\ columns\ of\ \hat{Q}_n$
            $k = 1$
            $y_n = \hat{Q}_{n-1}^T x_n;$
            $\hat{y}_n = \Delta(y_n)$
        end
    end
    $r_{opt} = k - 1$
    $\hat{y}_n(r_{opt} + 1 : N) = 0$
    $transmit\ \hat{y}_n, r_{opt}\ and\ side\ information\ to$
        $receiver$
    $F = \gamma\hat{\Lambda}_{n-1} + \hat{y}_n\hat{y}_n^T$
    $solve\ F\hat{Q}_n = \hat{Q}_n\hat{\Lambda}_n$
end
	$\hat{Q}_0 = I_N$
	$for\ n = 1, 2, \ldots$
	    $wait\ for\ \hat{y}_n,\ r_{opt},\ and\ side$
	        $information$
	    $\hat{x}_n = \hat{Q}_{n-1}\hat{y}_n$
	    $F = \gamma\hat{\Lambda}_{n-1} + \hat{y}_n\hat{y}_n^T$
	    $solve\ F\hat{Q}_n = \hat{Q}_n\hat{\Lambda}_n$
	end

Table 2: Algorithm AKLT2. This algorithm computes the entire set of eigenvectors and hence does not require a search direction.

inaccurate estimates of the signal. It is therefore desirable to have a mechanism which adjusts the signal subspace dimension to accommodate sudden changes in the signal subspace. By estimating the entire set of eigenvectors, we eliminate the need for finding a good search direction since the search space is the entire N-dimensional Euclidean vector space. To find the signal subspace dimension, the transmitter can measure the mean-squared error-like quantity $\rho \equiv \|x_n - \hat{x}_n\|^2$ (or a normalized version thereof). The signal subspace dimension can then be increased until ρ is below a preset threshold, MSE_{max}. The transmitter then sends the receiver the KLT coefficients as well as side information consisting of the number of KLT coefficients being transmitted (r_{opt}). The complete algorithm (AKLT2) is listed in Table 2. Algorithm AKLT2 can be extended one step further. Rather than updating the reduced autocorrelation matrix F with the KLT coefficients, we can simply update the sample autocorrelation matrix with the reconstructed signal frame, $\hat{x}_n$. Moreover, instead of updating $\hat{R}_n$ on a frame-by-frame basis, we can update it on a sample by sample basis by concatenating $\hat{x}_{n-1}$ with $\hat{x}_n$. This enables a more accurate estimate of the sample autocorrelation matrix since more signal vectors are used in its computation. The resulting algorithm, called AKLT3, is listed in Table 3. Interestingly, if the noise present in $\hat{x}_n$ is white, the eigenvectors of $\hat{R}_n$ will not be affected since the diagonal noise autocorrelation matrix does not alter the eigenvectors of the noise-free autocorrelation matrix.

4 Experiments

The following signal was generated

$$x(n) = \begin{cases} cos(0.35\pi n) + cos(0.78\pi n + 0.35\pi), & n = 1, \ldots, 999 \\ cos(0.6\pi n) + cos(0.8\pi n + 0.35\pi), & n = 1000, \ldots, 2,000 \end{cases} \quad (6)$$

sender	*receiver*
$\hat{Q}_0 = I_N$ $for\ n = 1, 2, \ldots$ $\quad y_n = \hat{Q}_{n-1}^T x_n$ $\quad find\ bit\ allocation\ using\ \hat{\Lambda}_{n-1}$ $\quad \hat{y}_n = \Delta\left(y_n\right)$ $\quad \rho = 1, k = 1, b = 0$ $\quad while\ \rho\ >\ MSE_{max}\ AND\ b < 2$ $\quad\quad \hat{x}_n = \hat{Q}_{n-1}(:, 1:k)\hat{y}_n(1:k);$ $\quad\quad \rho = \lVert \hat{x}_n - x_n \rVert^2$ $\quad\quad k = k + 1;$ $\quad\quad if\ k = N + 1\ AND\ \rho > MSE_{max}$ $\quad\quad\quad use\ bit\ allocation\ plan\ B$ $\quad\quad\quad b = b + 1$ $\quad\quad\quad k = 1$ $\quad\quad\quad y_n = \hat{Q}_{n-1}^T x_n;$ $\quad\quad\quad \hat{y}_n = \Delta\left(y_n\right)$ $\quad\quad end$ $\quad end$ $\quad if\ b \neq 2,\ then\ r_{opt} = k - 1$ $\quad if\ b = 2,\ then\ r_{opt} = N$ $\quad \hat{y}_n(r_{opt} + 1 : N) = 0$ $\quad transmit\ \hat{y}_n, r_{opt}\ and\ b\ to\ receiver$ $\quad w_n = \left[\ \hat{x}_{n-1}^T \quad \hat{x}_n^T\ \right]^T$ $\quad \hat{R}_{n-1,0} = \hat{R}_{n-1}$ $\quad for\ m = 1 : N,$ $\quad\quad v = w_n(m + 1 : m + N)$ $\quad\quad \hat{R}_{n-1,m} = \gamma \hat{R}_{n-1,m-1} + vv^T$ $\quad end$ $\quad \hat{R}_n = \hat{R}_{n-1,N}$ $\quad solve\ \hat{R}_n \hat{Q}_n = \hat{Q}_n \hat{\Lambda}_n$ end	$\hat{Q}_0 = I_N$ $for\ n = 1, 2, \ldots$ $\quad find\ bit\ allocation\ using\ \hat{\Lambda}_{n-1}$ $\quad wait\ for\ \hat{y}_n,\ r_{opt},\ and\ b$ $\quad if\ b \neq 0,\ use\ bit\ allocation\ Plan\ B$ $\quad \hat{x}_n = \hat{Q}_{n-1}\hat{y}_n$ $\quad w_n = \left[\ \hat{x}_{n-1}^T \quad \hat{x}_n^T\ \right]^T$ $\quad \hat{R}_{n-1,0} = \hat{R}_{n-1}$ $\quad for\ m = 1 : N,$ $\quad\quad v = w_n(m + 1 : m + N)$ $\quad\quad \hat{R}_{n-1,m} = \gamma \hat{R}_{n-1,m-1} + vv^T$ $\quad end$ $\quad \hat{R}_n = \hat{R}_{n-1,N}$ $\quad solve\ \hat{R}_n \hat{Q}_n = \hat{Q}_n \hat{\Lambda}_n$ end

Table 3: Algorithm AKLT3. This algorithm uses the reconstructed signal frame to compute the sample autocorrelation matrix from which the KLT basis functions are extracted. A typical frame consists of b (1 bit) r_{opt} ($log_2 N$ bits), and the r_{opt} KLT coefficients (the number of bits required for each KLT coefficient is variable, and depends on the bit allocation method). If the bit allocation for each coefficient is computed from the sample autocorrelation matrix eigenvalues, then it needn't be transmitted since the receiver can either compute it ($b = 0$) or use a pre-determined allocation (*Plan B*, $b = 1$).

Algorithm AKLT1 was applied to this signal with $r = 4$ and $\gamma = 0.8$. The search direction v_n was set to a zero-mean Gaussian white noise vector. To measure the algorithm's performance the *a priori* mean squared error between the original and reconstructed data frame was estimated as $\epsilon^o(n) = \|x_n - \hat{Q}_{n-1}\hat{Q}_{n-1}^T x_n\|^2$ where $\| \quad \|$ is the standard vector 2-norm. We note that this is a more valid measure than *a posteriori* error since in a compression scenario, the quantity being transmitted is $\hat{Q}_{n-1}^T x_n$. Figure 1 shows $\epsilon^o(n)$ for different values of N. As predicted, the convergence speed increases with decreasing N. The experiment was then repeated using algorithm AKLT2, using $MSE_{max} = 10^{-5}$. Since algorithm AKLT2 tracks the eigenvectors of $\hat{R}_n$ exactly, and since the mean squared error between $\hat{x}_n$ and x_n is always minimized by adjusting the signal subspace dimension, we chose as a measure of performance the lowest signal subspace dimension, r_{opt} for which the normalized mean squared error ρ for any given signal frame did not exceed MSE_{max}. The resulting plots for $N = 16, 32$, and 64 are shown in Fig. 1. It can be seen that since the signal consists of two sinusoids, the signal subspace dimension is 4, consequently, the algorithm has converged when $r_{opt} = 4$, no quantization was done in this experiment. Next we applied algorithm AKLT3 using quantization. A different quantizer was used for each KLT coefficient. The bit allocation rule we used was the standard rule for minimizing mean-squared error

$$b_k = B + 0.5log_2 \frac{[\hat{\Lambda}_{n-1}]_k}{\prod_{j=1}^{r_{opt}}[\hat{\Lambda}_{n-1}]_k}, \quad k = 1, \ldots, r_{opt} \tag{7}$$

where b_k is the number of bits allocated to the k^{th} KLT coefficient, B is a constant, and $[\hat{\Lambda}_{n-1}]_k$ is the k^{th} eigenvalue [1]. The maximum allowable input for each quantizer was set to $4[\hat{\Lambda}_{n-1}]_k^{1/2}$. Bit allocation for *Plan B* involved setting all $b_k = 8$, this was required if the initial MSE_{max} threshold wasn't met. If, after going to *Plan B*, the threshold still wasn't met, the reconstructed signal was accepted as is. By adjusting MSE_{max}, one can obtain different bit rates. In this experiment, we used the utterance "the pipe began to rust while new" as the original signal. Setting MSE_{max} to 0.01 led to an average bit rate of about 22 kilobits per second (kb/s); the reconstructed signal sounded virtually identical to the original. Increasing MSE_{max} to 0.03 produced a bit rate of 17 kb/s, and led to a small amount of degradation. Figure 2 shows spectrograms of the original, and the 22 kb/s and 17 kb/s reconstructed signals; clicking on the spectrograms will play the corresponding audio file.

5 Summary

Several algorithms for blindly tracking the KLT basis vectors using only the KLT coefficients and a minimal amount of side information were described. The first algorithm is a random search method which blindly tracks a low-dimensional signal subspace. The second algorithm updates the KLT basis functions based on the reconstructed signal vectors, which, if close to the original signal, yield accurate eigenvectors. The algorithms were demonstrated on a sinusoidal signal and on a speech signal. Good speech reproduction was obtained at bit rates in the broadband range.

References

[1] N. S. Jayant and P. Noll, *Digital Coding of Waveforms.* Prentice Hall, 1984.

[2] I. Karasolo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Acoust. , Speech, Signal Processing*, vol. 34, pp. 8–12, Feb. 1986.

[3] C. E. Davila, "Blind adaptive estimation of KLT basis vectors," *(submitted to) IEEE Transactions on Signal Processing*, Dec. 1999.

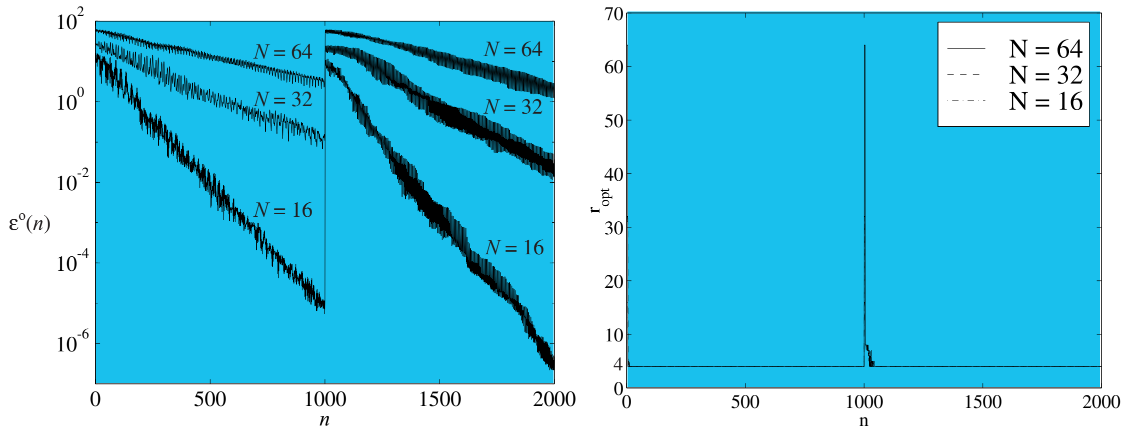

Figure 1: Algorithm AKLT1 mean squared error for sinusoid reconstruction for different frame lengths N using a white noise search direction (left) and signal subspace dimension required to reduce the mean squared error below MSE_{max} for algorithm AKLT2 (right).

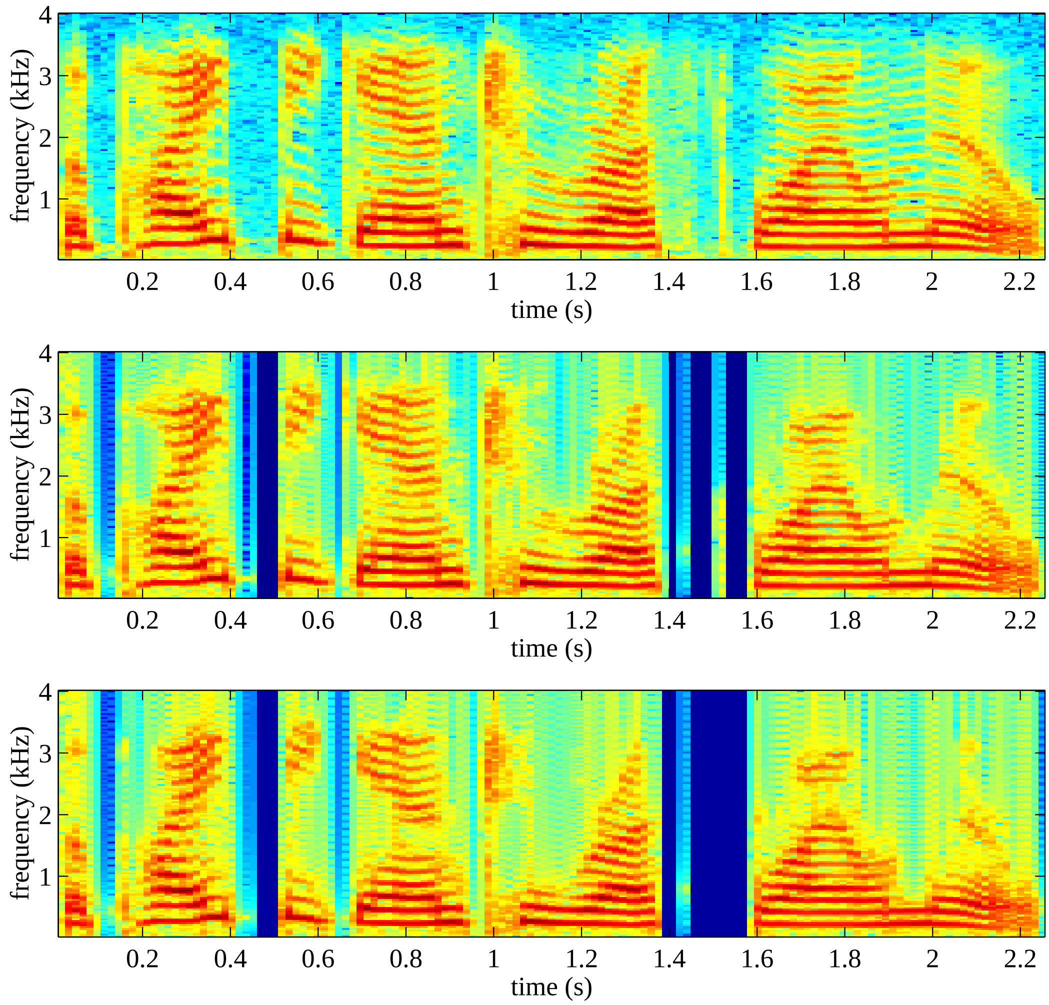

Figure 2: Spectrogram of original signal (top), 22 kb/s reconstructed signal (center), and 17 kb/s reconstructed signal (bottom) using algorithm AKLT3. Click on each spectrogram to listen to audio.