A Fast Derivation of Karhunen-Loève Transform Kernel for First-Order Autoregressive Discrete Process

Onur Yilmaz, Mustafa U. Torun, and Ali N. Akansu

New Jersey Institute of Technology
Department of Electrical and Computer Engineering
University Heights Newark, NJ 07102 USA
onur.yilmaz@njit.edu
mustafa.torun@njit.edu
akansu@njit.edu

Abstract

Karhunen-Loève Transform (KLT), also called principal component analysis (PCA) or factor analysis, based signal processing methods have been successfully used in applications spanning from eigenfiltering to recommending systems. KLT is a signal dependent transform and comprised of three major steps where each has its own computational requirement. Namely, statistical measurement of random data is performed to populate its covariance matrix. Then, eigenvectors (eigenmatrix) and eigenvalues are calculated for the given covariance matrix. Last, incoming random data vector is mapped onto the eigenspace (subspace) by using the calculated eigenmatrix. The recently developed method by Torun and Akansu offers an efficient derivation of the explicit eigenmatrix for the covariance matrix of first-order autoregressive, AR(1), discrete stochastic process. It is the second step of the eigenanalysis implementation as summarized in the paper. Its computational complexity is investigated and compared with the currently used techniques. It is shown that the new method significantly outperforms the others, in particular, for very large matrix sizes that are common in big data applications.

Keywords Eigendecomposition, factor analysis, principal component analysis (PCA), covariance analysis, first-order autoregressive AR(1) discrete process, explicit Karhunen-Loève Transform (KLT) kernel.

1. Introduction

Karhunen-Loève Transform (KLT) has been a popular mathematical method used in a variety of disciplines including image processing, finance, and social media analytics [1, 2]. Although the KLT has been known as the optimal block transform for a given signal source, it has a prohibitively high cost of implementation, particularly in real-time applications. There are numerical methods to derive the KLT basis functions (eigenvectors) for a given covariance matrix such as Jacobi [3–5], QR factorization (QR) [5, 6], and divide and conquer (D&Q) [5, 7] that still require sizable computational resources for large dimensional data. Therefore, signal independent (fixed) transforms, like the discrete Fourier transform (DFT) and discrete cosine transform (DCT), have been good approximations to KLT for certain signal types in practice, and used in a variety of technologies and systems [1].

Eigenfiltering of measurement noise and dimension reduction of big and sparse data are the two leading KLT applications that drive the current research activity on eigendecomposition [2, 8]. Although there was some progress on closed-form expressions for the KLT kernel of AR(1) process, its derivation and implementation is of a major concern. A few closed-form kernel expressions for certain processes have been reported in the literature [9–11].

First-order autoregressive, AR(1), discrete random process is considered as a coarse approximation to many natural signal sources. Recently, an efficient method to derive explicit KLT kernel for AR(1) process was developed [12]. This paper investigates the computational performance of the new kernel derivation method for large dimensions that may benefit big data applications, and compares with D&Q under the same test conditions.

KLT is comprised of three major steps where each has its own computational requirement. Namely, a. correlation measurements of raw data to populate covariance matrix; b. derivation of KLT matrix through eigendecomposition of covariance matrix; c. implementing KLT transform for a given data vector. This paper investigates computational cost of the fast KLT kernel generation method derived in [12] for step b here.

The paper is structured as follows. The eigenanalysis of AR(1) process with closed-form KLT kernel expression requiring the root locations of a transcendental tangent equation is summarized in Sec. II. A fast method proposed in [12] to derive explicit KLT kernel for AR(1) discrete process, FD-AR(1), is highlighted in Sec. III. In Sec. IV, we compare algorithm time complexity and computation time of KLT, derived by using the proposed kernel derivation method, and D&Q. Moreover, we emphasize implementation advantages of the proposed method in this section. The concluding remarks are given in the last section of the paper.

2. Eigenanalysis of AR(1) Discrete Process

2.1 AR(1) Signal Model

Autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) are widely used mathematical models to represent real-world signal sources. In particular, AR source models, also called all-pole models, have been successfully used in several areas including economics, weather forecasting, speech processing applications for decades. AR source model with order one, AR(1), is a first approximation to many natural signals. AR(1) signal is generated through the regression formula as written [1]

$$x\left(n\right)=\rho x\left(n-1\right)+\xi\left(n\right) \tag{1}$$

where $\xi(n)$ is a white noise sequence with zero-mean, i.e.

$$E\left\{\xi(n)\xi(n+k)\right\}=\sigma_{\xi}^{2}\delta_{k},\ E\{\xi(n)\}=0 \tag{2}$$

The first-order correlation coefficient, ρ, is real in the range of $-1<\rho<1$, and the variance of $x(n)$ is given as follows

$$\sigma_{x}^{2}=\left(1-\rho^{2}\right)^{-1}\sigma_{\xi}^{2} \tag{3}$$

Autocorrelation sequence of $x\left(n\right)$ is expressed as

$$R_{xx}(k)=\sigma_{x}^{2}\rho^{|k|};\ k=0,\pm1,\pm2,\dots \tag{4}$$

The resulting Toeplitz correlation matrix of size $N\times N$ is defined as

$$\mathbf{R}_{x}=\sigma_{x}^{2}\begin{bmatrix}1 & \rho & \cdots & \rho^{N-1}\\ \rho & 1 & \cdots & \rho^{N-2}\\ \vdots & \vdots & \ddots & \vdots\\ \rho^{N-1} & \rho^{N-2} & \cdots & 1\end{bmatrix} \tag{5}$$

The popularity of modeling signals as AR(1) process is due to its analytical simplicity and ease of implementation. We will summarize below a recently introduced fast eigenanalysis method for AR(1) discrete process [12]. Then, we will investigate its computational requirements and compare with the currently available techniques.

2.2 Eigendecomposition of AR(1) Discrete Process

An eigenvalue λ and an eigenvector $\boldsymbol{\phi}$ of a matrix $\boldsymbol{\Phi}$ with size $N\times N$ must satisfy the equation [1, 5, 10]

$$\begin{aligned}\mathbf{R}_{x}\boldsymbol{\phi}&=\lambda\boldsymbol{\phi}\\ \mathbf{R}_{x}\boldsymbol{\phi}-\lambda\mathbf{I}\boldsymbol{\phi}&=\left(\mathbf{R}_{x}-\lambda\mathbf{I}\right)\boldsymbol{\phi}=0\end{aligned} \tag{6}$$

such that $(\mathbf{R}_{x}-\lambda\mathbf{I})$ is singular. Namely,

$$\det\left(\mathbf{R}_{x}-\lambda\mathbf{I}\right)=0 \tag{7}$$

Since $\mathbf{R}_{x}$ is a real and symmetric matrix, its eigenvectors with different eigenvalues are linearly independent. Thus, this determinant is a polynomial in λ of degree N, (7) has N roots and (6) has N solutions for $\boldsymbol{\phi}$ that result in eigenpair set $\{\lambda_{k},\boldsymbol{\phi}_{k}\}$ where $0\leq k\leq N-1$. Therefore, the eigendecomposition of $\mathbf{R}_{x}$ with distinct eigenvectors is written as follows

$$\mathbf{R}_{x}=\mathbf{A}_{KLT}^{*\mathrm{T}}\boldsymbol{\Lambda}\mathbf{A}_{KLT}=\sum_{k=0}^{N-1}\lambda_{k}\boldsymbol{\phi}_{k}\boldsymbol{\phi}_{k}^{*\mathrm{T}} \tag{8}$$

where $\boldsymbol{\Lambda}=diag\left(\lambda_{k}\right);\ k=0,1,\dots,N-1$, and k*th* column of $\mathbf{A}_{KLT}^{*\mathrm{T}}$ matrix is the k*th* eigenvector $\boldsymbol{\phi}_{k}$ of $\mathbf{R}_{x}$ with the corresponding eigenvalue λ_{k}.

Note that $\left\{\lambda_{k}=\sigma_{k}^{2}\right\}\forall k$, for the given $\mathbf{R}_{x}$ where σ_{k}^{2} is the variance of the k*th* transform coefficient, θ_{k}. The eigenvalues of $\mathbf{R}_{x}$ for AR(1) process defined in (5) are derived to be in the closed-form [11]

$$\sigma_{k}^{2}=\lambda_{k}=\frac{1-\rho^{2}}{1-2\rho\cos\left(\omega_{k}\right)+\rho^{2}};\ 0\leq k\leq N-1 \tag{9}$$

where $\{\omega_{k}\}$ are the positive roots of the following transcendental tangent equation

$$\tan\left(N\omega\right)=-\frac{\left(1-\rho^{2}\right)\sin\left(\omega\right)}{\cos\left(\omega\right)-2\rho+\rho^{2}\cos\left(\omega\right)} \tag{10}$$

that is equivalent to [12]

$$\left[\tan\left(\omega\frac{N}{2}\right)+\gamma\tan\left(\frac{\omega}{2}\right)\right]\left[\tan\left(\omega\frac{N}{2}\right)-\frac{1}{\gamma}\cot\left(\frac{\omega}{2}\right)\right]=0 \tag{11}$$

where $\gamma=(1+\rho)/(1-\rho)$ and the resulting KLT matrix of size $N\times N$ is expressed in the explicit kernel as [11]

$$\mathbf{A}_{KLT}=\left[A\left(k,n\right)\right]=c_{k}\sin\left[\omega_{k}\left(n-\frac{N-1}{2}\right)+\frac{(k+1)\,\pi}{2}\right] \tag{12}$$

$$c_{k}=\left(\frac{2}{N+\lambda_{k}}\right)^{1/2},\ 0\leq k,n\leq N-1 \tag{13}$$

3. Fast Derivation of Explicit KLT Kernel for AR(1) Process

In order to derive an explicit expression for the roots of the transcendental equation that are required in the definition of the discrete KLT kernel given in (12), we need to calculate the first $N/2$ positive roots of two transcendental equations as given [11, 12]

$$\tan\left(\omega\frac{N}{2}\right)=-\gamma\tan\left(\frac{\omega}{2}\right) \tag{14}$$

$$\tan\left(\omega\frac{N}{2}\right)=\frac{1}{\gamma}\cot\left(\frac{\omega}{2}\right) \tag{15}$$

where N is the transform size, $\gamma=(1+\rho)/(1-\rho)$ and ρ is the first-order correlation coefficient for AR(1) discrete process. Roots of (14) and (15) correspond to the odd and even indexed eigenvalues and eigenvectors respectively. The equation to find the mth root is defined as [12]

$$\omega_{m}=h_{m}+R_{m}\left[\frac{\int_{0}^{2\pi}w_{m}(\theta)e^{j2\theta}d\theta}{\int_{0}^{2\pi}w_{m}(\theta)e^{j\theta}d\theta}\right] \tag{16}$$

For the roots of (14) following parameters can be used in (16) [12]

$$\begin{aligned}R_{m}&=\begin{cases}\pi/2N & m\leq2\\ \pi/N & m>2\end{cases}\\ h_{m}&=\left(m-1/4\right)\left(2\pi/N\right)\ \ 1\leq m\leq N/2\end{aligned} \tag{17}$$

$$w_{m}(\theta)=\frac{1}{\tan\left[\left(h_{m}+R_{m}e^{j\theta}\right)\frac{N}{2}\right]+\gamma\tan\left[\left(h_{m}+R_{m}e^{j\theta}\right)\frac{1}{2}\right]} \tag{18}$$

Similarly, for the roots of (15) following parameters can be used in (16) [12]

$$\begin{aligned}R_{m}&=\pi/N\\ h_{m}&=\begin{cases}\left(m-1/2\right)\left(2\pi/N\right) & m\leq2\\ \left(m-1\right)\left(2\pi/N\right) & m>2\end{cases}\end{aligned} \tag{19}$$

$$w_m(\theta) = \frac{1}{\tan\left[\left(h_m + R_m e^{j\theta}\right)\frac{N}{2}\right] - \frac{1}{\gamma}\cot\left[\left(h_m + R_m e^{j\theta}\right)\frac{1}{2}\right]} \tag{20}$$

The term in the brackets in (16) is equal to ratio of the second Fourier series coefficient over the first one [12]. Thus, *m*th root can be easily calculated by only using the first and second basis vectors of transformation matrix of DFT. After the ω_m is calculated, $\mathbf{A}_{KLT}$ can be calculated by plugging ω_m into (12) [12].

The steps of the algorithm proposed in [12] to derive an explicit KLT kernel of dimension N for an arbitrary discrete data set modeled by an AR(1) source are summarized as follows.

1. The first-order correlation coefficient $\rho = R_{xx}(1)/R_{xx}(0) = E\left\{x(n)x(n+1)\right\}/E\left\{x(n)x(n)\right\}$ of AR(1) model for the given discrete data set $\{x(n)\}$ is estimated where $-1 < \rho < 1$.
2. The positive roots $\{\omega_k\}$ of the polynomial given in (11) are calculated by substitution of (18) and (20) into (16) for odd and even values of k, respectively, where

   $$m = \begin{cases} k/2+1 & k \text{ even} \\ (k+1)/2 & k \text{ odd} \end{cases} \tag{21}$$

   These roots are efficiently calculated by using DFT in the root finding algorithm detailed in [12].
3. Use the values of ρ and $\{\omega_k\}$ found from (16) in (9) and (13) to calculate the eigenvalues and eigenvectors, respectively. Hence the KLT matrix $\mathbf{A}_{KLT}$ is defined.

Main contribution of this paper is to quantify the computational load of this new derivation method for explicit KLT kernel for AR(1) discrete process and compare with the current state-of-the-art.

4. Performance Evaluation

In this section, we evaluate the computational performance of the proposed method and compare with the widely used numerical algorithms in terms of algorithm time complexity, numerical error, and computation time required. We performed the tests on a six core (a total of twelve cores with hyper-threading) Intel® Core™ i7-3960X CPU @ 3.20GHz with 24 GB RAM machine running on Windows and using MATLAB™.

4.1 Algorithm Time Complexity

In this section, we compare the time complexity of explicit kernel derivation method based on AR(1) approximation, FD-AR(1), and the widely used numerical methods. Jacobi method [5] is the most stable algorithm among the numerical methods [4]. However, since it is an iterative method, it is hard to quantify its time complexity. For only one sweep, time complexity of the algorithm is $O(N^3)$ where N is the matrix size. The time complexity of the algorithm increases according to the required number of sweeps. QR and D&Q are two other popular methods employed in many applications [5]. Time complexity for both is $O(N^3)$. On the other hand, the time complexity of FD-AR(1) is $O(NL)$, where L is the DFT size used in the root finding step. Therefore, for large dimensions, that are common in big data applications, FD-AR(1) offers significant complexity reductions compared to the existing KLT kernel derivation methods as tabulated in Table 1.

Method	Complexity
Jacobi [3]	$O(N^3)$
QR [13]	$O(N^3)$
D&Q [7]	$O(N^3)$
Fast Derivation (FD-AR(1)) [12]	$O(NL)$

Table 1: Algorithm Time Complexity Comparison

4.2 Numerical Error of DFT-based Root Finding in FD-AR(1)

In this section, numerical error ϵ of FD-AR(1) method due to the DFT approximation to the Fourier series coefficients used in the root finding step is investigated [12]. An error metric that quantifies the orthogonality imperfectness of the KLT matrix is defined as follows

$$\epsilon = \left\| \mathbf{I} - \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} \right\|_2 \tag{22}$$

where $\|\cdot\|_2$ is 2-norm, $\mathbf{\Phi}$ and $\mathbf{I}$ are the KLT and identity matrices, respectively. Table 2a displays the measured error ϵ for various L and N with $\rho = 0.95$. It is observed from the Table 2a that when the DFT size L is increased, the numerical error decreases as expected. Table 2b displays the measured error ϵ for various L and $N = 5,000$ with $\rho = 0.95$. Larger L values are selected for this table to keep ϵ low. Similarly, Fig.1 displays the measured error ϵ with respect to ρ values for $N = 1,000$. This figure shows that ϵ is significantly lower for highly correlated AR(1) signals. It is concluded that one should define a proper value of L for the desired N and correlation range of interest that will keep ϵ low. In contrast, D&Q has an error ϵ in the order of 10^{-14} that is independent of ρ and N due to its iterative nature.

N	L=128	L=256	L=512	L=1,024
200	1.4×10^{-9}	1.0×10^{-11}	4.4×10^{-12}	1.8×10^{-12}
400	5.1×10^{-6}	1.0×10^{-10}	2.3×10^{-11}	1.9×10^{-11}
600	1.3×10^{-4}	7.1×10^{-8}	3.7×10^{-11}	2.2×10^{-11}
800	8.1×10^{-4}	2.4×10^{-6}	7.4×10^{-11}	3.2×10^{-12}
1,000	2.5×10^{-3}	2.2×10^{-5}	1.8×10^{-9}	1.0×10^{-10}

(a)

N	L=1,024	L=2,048	L=4,096	L=8,192
5,000	1.0×10^{-4}	4.0×10^{-8}	6.5×10^{-10}	1.6×10^{-10}

(b)

Table 2: (a) Numerical error ϵ of DFT-based root finding in FD-AR(1) for various L and N with $\rho = 0.95$. (b) Numerical error ϵ of DFT-based root finding in FD-AR(1) for various L and $N = 5,000$ with $\rho = 0.95$.

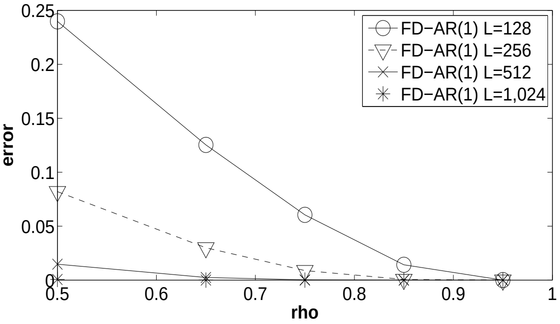

Figure 1: Numerical error ϵ with respect to ρ (rho) and L for $N = 1,000$.

4.3 Computation Time

In this section, computation times to generate KLT matrices obtained by using D&Q and FD-AR(1) methods are measured and compared. The measurements include the kernel derivation for the case of $\rho = 0.95$. Computation of $\mathbf{R}_x$ of (5) is also included in the D&Q measurements since it is required. All computing experiments reported are conducted by using MATLAB™.

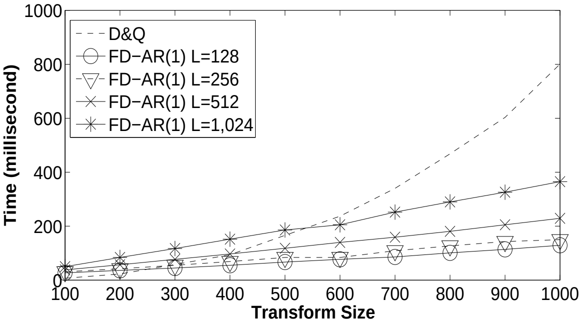

Figure 2: Computation times to generate KLT matrices obtained by D&Q and FD-AR(1) for various L and up to $N = 1,000$.

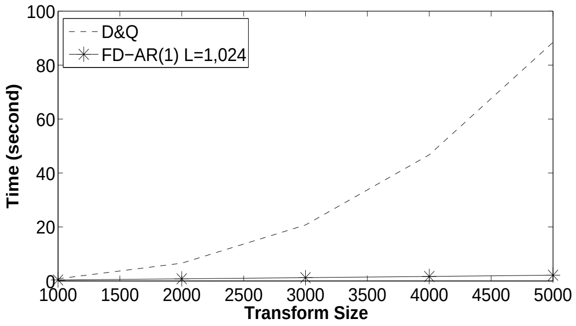

Figure 3: Computation times to generate KLT matrices obtained by D&Q and FD-AR(1) for various L and up to $N = 5,000$.

Fig. 2 displays the computation times to generate KLT matrices generated by D&Q and FD-AR(1) for various L values and up to $N = 1,000$. For $N < 250$, D&Q is slightly faster than FD-AR(1). On the other hand, FD-AR(1) begins to show its merit for $N > 250$. It can be observed from Fig. 3 that FD-AR(1) significantly outperforms the D&Q for $N \geq 1,000$.

N	L	Time(second)	ϵ
$35,000$	$16,384$	68.253	9.0×10^{-9}

Table 3: Computation time to generate KLT matrix obtained by FD-AR(1) and the resulting numerical error ϵ for $L = 16,384$ and $N = 35,000$ in the case of $\rho = 0.95$.

It is displayed in Table 3 that FD-AR(1) method with $L = 16,384$ derives the size $N = 35,000$ KLT matrix in 68.25 seconds with $\epsilon = 9.0 \times 10^{-9}$. The dimension $N = 35,000$ is the maximum limit for the machine used in the tests due to memory limitation. Note that it is not possible to run D&Q algorithm for $N = 35,000$ on the same machine. The matrix size N may be increased by implementing the algorithm to be run on a cluster of machines via Hadoop [14] that is a widely used tool for big data applications.

5. Conclusions

An efficient method to derive explicit KLT kernel for AR(1) discrete process, FD-AR(1), was introduced in [12]. This method offers attractive computational performance where the big data applications may benefit from. In this paper, we highlighted the merit of FD-AR(1) by comparing its performance with the currently existing methods in terms of algorithm time complexity, error (orthonormality imperfectness), and computation time. FD-AR(1) is shown to be significantly superior for large matrix sizes. The kernel derivation error results suggest that one should define a proper value of L for the desired N and correlation range of interest in order to keep ϵ low.

KLT is almost impractical for real-time (or near real-time) applications due to its computational cost. We validate FD-AR(1) as a feasible signal processing tool for eigenfiltering and dimension reduction of big data applications. Moreover, other discrete processes like higher order AR, MA and ARMA may be approximated by using AR(1) [15] FD-AR(1) for real-time data intensive applications employing eigenanalysis. The authors are currently studying FD-AR(1) implementations on graphics processing unit (GPU) and field programmable gate arrays (FPGA) for emerging high performance DSP applications.

References

[1] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets.* Academic Press, Inc., 1992.

[2] D. Kim and B.-J. Yum, "Collaborative filtering based on iterative principal component analysis," *Expert Syst. Appl.*, vol. 28, pp. 823–830, May 2005.

[3] C. G. J. Jacobi, "Über ein leichtes verfahren, die in der theorie der säkularstörungen vorkommenden gleichungen numerisch aufzulösen," *Crelle's Journal*, vol. 30, pp. 51 – 94, 1846.

[4] J. Demmel and K. Veselic, "Jacobi's method is more accurate than QR," *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 1204 – 1245, 1992.

[5] G. H. Golub and C. F. V. Loan, *Matrix Computations.* Johns Hopkins University Press, 1996.

[6] Y. S. Nathan Halko, Per-Gunnar Martinsson and M. Tygert, "An algorithm for the principal component analysis of large data sets," *SIAM Journal of Scientific Computing*, vol. 33, pp. 2580–2594, Jan. 2011.

[7] J. D. Rutter, "A serial implementation of cuppen's divide and conquer algorithm," tech. rep., Berkeley, CA, USA, 1991.

[8] A. Cuzzocrea, I.-Y. Song, and K. C. Davis, "Analytics over large-scale multidimensional data: the Big data revolution!," in *Proc. ACM 14th international Workshop on Data Warehousing and OLAP*, pp. 101–104, 2011.

[9] W. B. Davenport and W. L. Root, *An Introduction to the Theory of Random Signals and Noise.* McGraw-Hill, 1958.

[10] J. Wilkinson, *The Algebraic Eigenvalue Problem.* Oxford University Press, 1965.

[11] W. Ray and R. Driver, "Further decomposition of the Karhunen-Loève series representation of a stationary random process," *IEEE Trans. Information Theory*, vol. 16, pp. 663–668, Sept. 1970.

[12] M. U. Torun and A. N. Akansu, "An efficient method to derive explicit KLT kernel for first-order autoregressive discrete process," *Accepted to IEEE Trans. Signal Processing*, 2013.

[13] B. N. Parlett, "The QR algorithm," *Computing in Science and Eng.*, vol. 2, pp. 38–42, Jan. 2000.

[14] Apache Hadoop, *Apache Hadoop 2.0.3, http://hadoop.apache.org/*, 2013.

[15] S. Kay, *Modern Spectral Estimation: Theory and Application.* Prentice-Hall, 1988.