PERFORMANCE OF THE DISTRIBUTED KLT AND ITS APPROXIMATE IMPLEMENTATION

Mauricio Lara[1] *and Bernard Mulgrew*[2]

[1] Sección de Comunicaciones, Cinvestav-IPN, Av. IPN 2508, CP. 07360, Mexico City, Mexico.
[2] Institute for Digital Communications, University of Edinburgh, Edinburgh EH9 3JL, UK.
mlara@cinvestav.mx, B.Mulgrew@ed.ac.uk

ABSTRACT

The Karhunen-Loève Transform (KLT) is an important processing tool in data compression. Recently, a distributed version of the KLT was introduced together with an iterative algorithm for its implementation. Then, a recursive greedy algorithm for its approximate implementation was presented. Performance evaluations of both algorithms in the literature have been based on a few toy examples, which is not enough to show the intricacies of the distributed KLT. In this paper, a multi-terminal Markov-chain model is presented and used to evaluate the KLT algorithms using a large number of randomly generated covariance matrices. One of the findings is that the recursive greedy algorithm does not perform as well as predicted, and thus a variation of this algorithm is proposed which outperforms the original one. Finally, a promising application for both approximate algorithms is considered when the optimal distribution of the transmitted vectors among terminals is to be obtained.

Index Terms— Distributed compression, distributed transforms, distributed KLT, multi-terminal model.

1. INTRODUCTION

The Distributed Karhunen-Loève Transform (KLT) was introduced in [1], and it is called to play a major role in distributed compression and estimation [2,3]. Regrettably, the problem posed in [1] does not seem to accept a closed solution, and an iterative algorithm had to be proposed for its implementation. Afterwards, a greedy algorithm was put forward in [4] to approximately compute the distributed KLT with a reduction in complexity, and was found to achieve performance very close to that of the iterative algorithm. However, performance evaluations in [1] and [4] were carried out based solely on a few toy examples, and evaluations in more general scenarios are needed.

In this paper, a multi-terminal Markov-chain model is presented, and it is used to evaluate the distributed KLT algorithms with a large number of randomly generated covariance matrices. A variation of the greedy algorithm is proposed and shown to outperform the original algorithm at the expense of an increase in computation complexity. Also, an application for both the greedy algorithm and its modified version is investigated in the situation where the optimal distribution of the transmitted vectors among terminals is not known and has to be obtained.

M. Lara thanks The University of Edinburgh for the facilities to do this research and Conacyt, Mexico, for the financial support.

2. THE KARHUNEN-LOÈVE TRANSFORM

Let $\boldsymbol{x}$ be an $N \times 1$ vector of zero-mean, real-valued correlated random variables

$$\boldsymbol{x} = [x_1\, x_2 \cdots x_N]^T \tag{1}$$

with covariance matrix

$$\boldsymbol{\Sigma}_x = E\{\boldsymbol{x}\boldsymbol{x}^T\} \tag{2}$$

Through a linear transformation based on an $n \times N$ fat matrix $\boldsymbol{C}$, an $n \times 1$ approximation $\boldsymbol{y}$ of $\boldsymbol{x}$ is transmitted to a reconstruction terminal, where an $N \times n$ tall matrix $\boldsymbol{W}$ is used to get a linear estimate $\hat{\boldsymbol{x}}$ of the original vector $\boldsymbol{x}$.

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}\,; \qquad \hat{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{y} = \boldsymbol{W}\boldsymbol{C}\boldsymbol{x} \tag{3}$$

In a classical problem of linear estimation, the objective is to find matrices $\boldsymbol{C}$ and $\boldsymbol{W}$ so as to minimize the mean square error (MSE) D between vector $\boldsymbol{x}$ and its estimate $\hat{\boldsymbol{x}}$

$$D = E\{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2\} \tag{4}$$

Moreover, the problem can be reduced to that of finding $\boldsymbol{C}$, since given matrix $\boldsymbol{C}$, the reconstruction matrix $\boldsymbol{W}$ that minimizes the MSE can be found to be

$$\boldsymbol{W} = \boldsymbol{\Sigma}_x \boldsymbol{C}^T (\boldsymbol{C}\boldsymbol{\Sigma}_x \boldsymbol{C}^T)^{-1} \tag{5}$$

2.1. The joint KLT

The answer to the above classical estimation problem is well known [5]. Let $\boldsymbol{Q}_x$ be the $N \times N$ unitary matrix whose columns are the eigenvectors of $\boldsymbol{\Sigma}_x$, ordered by decreasing eigenvalues. Matrix $\boldsymbol{Q}_x^T$ is referred to as the Karhunen-Loève Transform (KLT) of $\boldsymbol{x}$. If $\boldsymbol{Q}_{x,n}$ represents the $N \times n$ matrix formed by the first n columns of $\boldsymbol{Q}_x$, then the solution to the above optimization problem is given by

$$\boldsymbol{C} = \boldsymbol{Q}_{x,n}^T \tag{6}$$

We will refer to (6) as the *joint* KLT of vector $\boldsymbol{x}$.

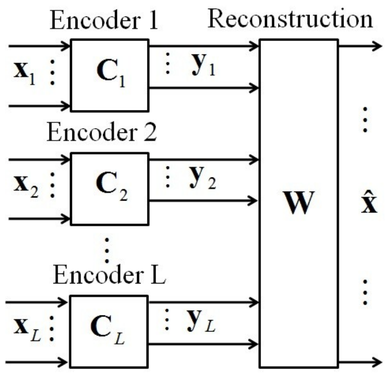

Fig. 1. The distributed KLT in a multi-terminal environment.

3. THE DISTRIBUTED KLT

Consider the distributed scenario shown in Fig. 1 involving L terminals. Terminal 1 observes vector $\boldsymbol{x}_1$, made up of the first M_1 components of vector $\boldsymbol{x}$; terminal 2 observes vector $\boldsymbol{x}_2$, made up of the following M_2 components of vector $\boldsymbol{x}$; and so on up to terminal L. Terminal i has to transmit a $k_i \times 1$ approximation $\boldsymbol{y}_i$ of vector $\boldsymbol{x}_i$ making use of a $k_i \times M_i$ encoding matrix $\boldsymbol{C}_i$, for $i = 1, \cdots, L$. The values $M_i, i = 1, \cdots, L$ add to N, and the values $k_i, i = 1, \cdots, L$ add to n, the dimension of the whole transmitted signal. Vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, and matrix $\boldsymbol{C}$ can be rewritten as

$$\boldsymbol{x} = [\boldsymbol{x}_1^T \, \boldsymbol{x}_2^T \cdots \boldsymbol{x}_L^T]^T \, ; \qquad \boldsymbol{y} = [\boldsymbol{y}_1^T \, \boldsymbol{y}_2^T \cdots \boldsymbol{y}_L^T]^T \tag{7}$$

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{C}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{C}_L \end{pmatrix} \tag{8}$$

In the receiving terminal, the L approximations $\boldsymbol{y}_i$ from the terminals are jointly used as the input to an $N \times n$ reconstruction matrix $\boldsymbol{W}$ to obtain a linear estimate $\hat{\boldsymbol{x}}$ of the complete vector $\boldsymbol{x}$. Again, the objective is to find the matrix $\boldsymbol{C}$ that minimizes the MSE D defined in (4). This time, however, matrix $\boldsymbol{C}$ is restricted to be block-diagonal as in (8), and no closed solution seems to exist to date in the literature. We will describe an iterative approach to compute the distributed KLT introduced in [1] as well as a recursive approximation presented in [4].

3.1. The marginal KLT

One simple, albeit suboptimal, solution is for each terminal to transmit a k_i-dimensional approximation $\boldsymbol{y}_i$ based on the standard KLT applied to its M_i-dimensional observation vector $\boldsymbol{x}_i$. This way, each terminal only needs to know the covariance matrix of its own observation vector $\boldsymbol{x}_i$, and its encoding matrix is computed as

$$\boldsymbol{C}_i = \boldsymbol{Q}_{x_i,k_i}^T \tag{9}$$

where $\boldsymbol{Q}_{x_i,k_i}^T$ comes from the first k_i rows of the KLT of vector $\boldsymbol{x}_i$. Matrix $\boldsymbol{C}$ is then formed according to (8). We call this solution the *marginal* KLT of vector $\boldsymbol{x}$.

3.2. The iterative local KLT

Let us first assume that we have only two terminals α and β. Terminal α observes vector $\boldsymbol{x}_\alpha$, made up of the first M components of $\boldsymbol{x}$, and transmits a $k \times 1$ approximation $\boldsymbol{y}_\alpha$. Terminal β observes vector $\boldsymbol{x}_\beta$, made up of the last $N - M$ components of vector $\boldsymbol{x}$, and transmits an $(n-k) \times 1$ approximation $\boldsymbol{y}_\beta$. Suppose that terminal β has decided on an appropriate $(n-k) \times (N-M)$ encoding matrix $\boldsymbol{C}_\beta$. The objective is to find the $k \times M$ encoding matrix $\boldsymbol{C}_\alpha$ of terminal α that minimizes the MSE in (4). The solution to this problem is given in closed form in [1], equations (12) to (18), and it is called the *local* KLT solution.

Also in [1], an iterative terminal-by-terminal approach is proposed for computing the distributed KLT based on the local KLT. First, the encoding matrices $\boldsymbol{C}_i, i = 1, \cdots, L$ are initialized arbitrarily. Then, in one iteration of the algorithm, for each terminal $j = 1, \cdots, L$, one at a time, the algorithm proceeds as follows. The input vector $\boldsymbol{x}_j$ and the encoding matrix $\boldsymbol{C}_j$ take the role of $\boldsymbol{x}_\alpha$ and $\boldsymbol{C}_\alpha$ of the local KLT, so that $\boldsymbol{x}_\alpha = \boldsymbol{x}_j$. The input vector to terminal β is formed by removing vector $\boldsymbol{x}_j$ from the complete input vector $\boldsymbol{x}$

$$\boldsymbol{x}_\beta = \left[\boldsymbol{x}_1^T \cdots \boldsymbol{x}_{j-1}^T \, \boldsymbol{x}_{j+1}^T \cdots \boldsymbol{x}_L^T\right]^T \tag{10}$$

while the encoding matrix $\boldsymbol{C}_\beta$ is formed by removing from matrix $\boldsymbol{C}$ in (8) the columns and rows of matrix $\boldsymbol{C}_j$. Next, $\boldsymbol{C}_\alpha$ is computed according to the local KLT explained above, and the encoding matrix $\boldsymbol{C}_j$ is updated as $\boldsymbol{C}_j = \boldsymbol{C}_\alpha$. The algorithm continues for a fixed number iterations or until certain criterion is met, for example, when the difference in the resulting MSE is less than a tolerance value ϵ. This will be called the *iterative local* KLT algorithm.

4. GREEDY-ALGORITHM KLT

4.1. The greedy-algorithm KLT

In [4] a greedy algorithm is presented as an approximation to the distributed KLT. Its main characteristic is that it finds all the required matrices $\boldsymbol{C}_i, i = 1, \cdots, L$ in n steps, one row of a particular $\boldsymbol{C}_i$ at a time. Starting with all encoding matrices empty, at each step, the algorithm determines the single terminal that, by adding another dimension to its compressed version, attains the largest reduction in the MSE. Only the encoding matrix of that terminal is updated in this step. Any row of a matrix $\boldsymbol{C}_i$ that has been obtained earlier is kept unchanged in the following steps of the algorithm. This means that, at some point, an optimization problem somewhat different from those tackled in [1] appears. Thus, additional equations are developed in [4] to deal with this optimization problem, and a recursive implementation of the algorithm is presented. Evaluation of this *greedy-algorithm* KLT in [4] with toy examples and particular conditions shows excellent approximation to the local iterative KLT.

4.2. The modified greedy KLT

During our simulations, we realized that the distributed KLT was very sensitive to any adjustments to the encoding matrices, and leaving unchanged the previously found rows of a matrix when other encoding matrices have already changed, as the greedy-algorithm KLT does, might impair the performance of the final result. Thus, we propose a variation of the greedy algorithm in [4] with the same philosophy. This is, the modified algorithm finds all required matrices $\boldsymbol{C}_i, i=1,\cdots,L$ in n steps, augmenting one row of a particular $\boldsymbol{C}_i$ at a time. The difference is that the rows of matrix $\boldsymbol{C}_i$ that have been obtained earlier are recomputed when testing new hypotheses; and this recomputed matrix is used for the winning candidate.

Thus, at the end of step m, matrix $\boldsymbol{C}$ has dimensions $m \times N$. Referring to Section 3.2, suppose that terminal i is under consideration, and that the $k_i^{(m)} \times M_i$ matrix $C_\alpha^{(m)}$ and the $(m-k_i^{(m)}) \times (N-M_i)$ matrix $C_\beta^{(m)}$ have been defined. At step $(m+1)$ the objective will be to optimally find a new $(k_i^{(m)}+1) \times M_i$ augmented matrix $C_\alpha^{(m+1)}$, provided that $k_i^{(m)} < k_i$. In doing so, while the greedy algorithm keeps fixed both matrix $C_\beta^{(m)}$ and sub-matrix $C_\alpha^{(m)}$ of $C_\alpha^{(m+1)}$, the modified algorithm keeps fixed only matrix $C_\beta^{(m)}$; this means that matrix $C_\alpha^{(m+1)}$ is wholly recomputed. This process is completed for each test terminal $i=1,\cdots,L$, and only the encoding matrix $C_\alpha^{(m+1)}$ of the terminal that delivers the largest reduction in the MSE is updated.

Both greedy algorithms find matrices $\boldsymbol{C}_i, i=1,\cdots,L$ in n steps. For each test terminal in the modified algorithm, matrix $C_\alpha^{(m+1)}$ can be computed based on equations (12) to (18) in [1], as for the iterative algorithm. Thus the computational load per iteration step increases cubically with N [4]. Since for the original greedy algorithm only the largest eigenvector is needed to test each hypothesis, the computational load can be made to increase just quadratically with N [4], which represents a significant reduction in the computational burden. On the other hand, we will show that the *modified greedy* KLT algorithm outperforms the greedy-algorithm KLT in the MSE.

4.3. Optimal allocation of transmitted signals

Let us consider the problem of how to best allocate the n transmitted signals among terminals in the context of the distributed KLT. The brute force approach would be to test the iterative local KLT for each possible combination, and select the one that provides the minimum MSE. However, when the number L of terminals and size M_i of the input vectors are not small, this number of combinations can be very large indeed. We study the possibility of exploiting the greedy or the modified greedy KLT algorithms to do this job: after all, they are naturally fit for this application.

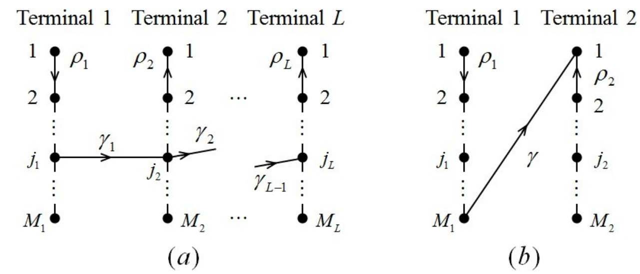

Fig. 2. Multi-terminal model based on Markov chains.

Accordingly, given n, the dimension of the transmitted signal, we let the greedy or modified greedy algorithms to increase the number $k_i^{(m)}$ of rows of the encoded vectors $\boldsymbol{C}_i$ without the restriction of being limited by a certain value other than $k_i^{(m)} \le M_i,\ i=1,2,\cdots L$ and $\sum k_i^{(m)} \le n$.

5. SIMULATION OF THE DISTRIBUTED KLT

5.1. The multi-terminal Markov-chain model

A random $N \times N$ covariance matrix can be generated as

$$\boldsymbol{\Sigma}_x = \boldsymbol{P}\boldsymbol{R}\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{R}^T\boldsymbol{P}^T \tag{11}$$

where $\boldsymbol{G}$ is an $N \times N$ matrix with elements drawn from a zero-mean uncorrelated Gaussian distribution with variance $1/N$, $\boldsymbol{R}$ is an $N \times N$ matrix with unit energy rows that defines the cross-correlation coefficients of the source, and $\boldsymbol{P}$ is an $N \times N$ diagonal matrix that defines the power of the individual components of the vector source. It is desirable to have a covariance model that can be parameterized with just a few variables. For matrix $\boldsymbol{R}$ we propose a multi-terminal Markov-chain model as shown in Fig. 2(a). The parameters are $\rho_i, i=1,\cdots,L$ that represents the correlation coefficient of the source in terminal i, and $\gamma_i, i=1,\cdots,L-1$ that represents the correlation coefficient between a given element j_i of the source vector in terminal i and a given element j_{i+1} of the vector in terminal $i+1$. For matrix $\boldsymbol{P}$ we propose a diagonal matrix with diagonal elements

$$\boldsymbol{P}_{i,i} = \sigma_{x,i} = \begin{cases} \lambda^{1/2}\alpha^{(i-1)/2}, & \alpha < 1 \\ 1, & \alpha = 1 \end{cases},\quad i=1,\cdots,N \tag{12}$$

where, in order to preserve the unit average variance of the whole source, λ is set as

$$\lambda = N\frac{1-\alpha}{1-\alpha^N},\quad \alpha < 1 \tag{13}$$

Note that the covariance matrices used in [1] and [4] are deterministic, while those used here are random. Thus, every element of Σ_x in (11) is a random variable whose mean value is given by the corresponding element of

$$\overline{\boldsymbol{\Sigma}}_x = E\{\boldsymbol{\Sigma}_x\} = \boldsymbol{P}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{P}^T \tag{14}$$

5.2. Simulation set-up

The number of terminals considered is $L=2$, the total number of input signals is $N=40$, and the total number of transmitted signals is, unless otherwise stated, $n=20$.

Scenario	ρ_1	ρ_2	γ	α	M_1	M_2
1	0	0	0	1	20	20
2	0.8	0.8	0	1	20	20
3	0.8	0.8	0.8	1	20	20
4	0	0.8	0.5	1	20	20
5	0	0	0	0.96594	20	20
6	0	0	0	1	25	15

Table 1. Definition of scenarios of the multi-terminal model.

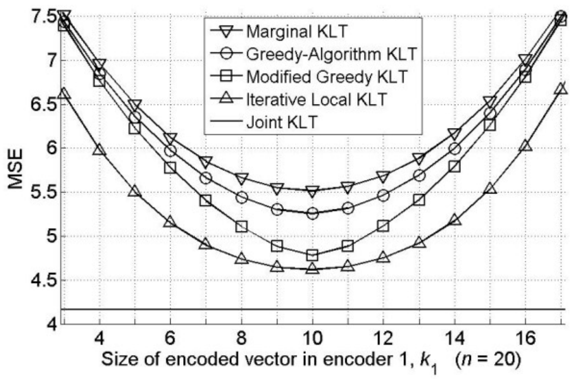

Fig. 3. MSE performance of different distributed KLT algorithms versus the size of encoded vector for scenario 1.

Using the above model, six different multi-terminal scenarios are defined in table 1. In all of them, the last component of terminal 1 is connected through the Markov chain to the first component of terminal 2, as shown in Fig. 2(b). The first three scenarios are symmetric, i.e. $\rho_1 = \rho_2$, $M_1 = M_2$ and $\alpha = 1$, and the last three scenarios are asymmetric. In scenario 5, the source in terminal 1 has more power that the source in terminal 2. For $N = 40$, the value of $\alpha = 0.96594$ results in the first 20 elements of $\boldsymbol{x}$ having double the average power of the last 20 elements. In scenario 6, the source in terminal 1 has also more power than that one in terminal 2, this time because it has more components of the input vector. The performance criterion is the MSE, and it is based on 500 realizations of the random matrix $\boldsymbol{G}$ in (11). Throughout the simulations, the iterative KLT algorithm is run with three different random initializations, and the run with minimum MSE is selected.

5.3. Simulation results

Simulations were carried out to compare the performance of the different distributed KLT algorithms. Figs. 3 to 6 show the results for scenarios 1, 3, 4 and 6. The results for scenarios 2 and 5 are not shown, but the performance in scenario 2 is very similar to that one in scenario 3. The performance in scenario 5 is somewhat similar to that one in scenario 6, but the MSEs are around 0.5 units lower and the MSEs of both greedy algorithms are always between those of the joint and marginal KLTs. As expected, the MSEs are lower when the source components are more correlated or have unequal distribution of power. We observe that for the

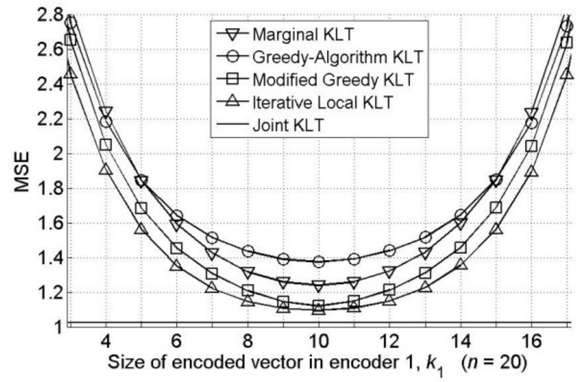

Fig. 4. MSE performance of different distributed KLT algorithms versus the size of encoded vector for scenario 3.

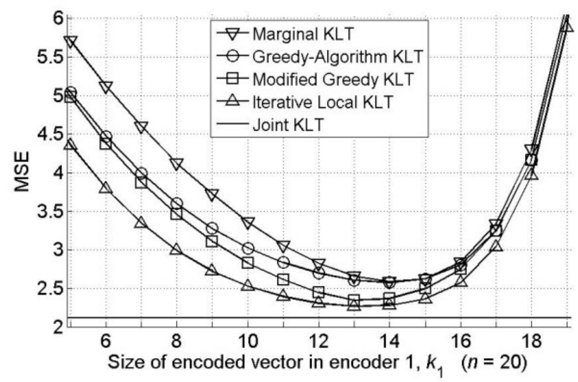

Fig. 5. MSE performance of different distributed KLT algorithms versus the size of encoded vector for scenario 4.

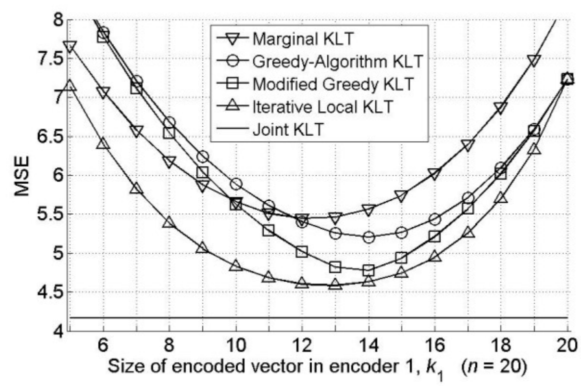

Fig. 6. MSE performance of different distributed KLT algorithms versus the size of encoded vector for scenario 6.

symmetrical settings (scenarios 1, 2 and 3), lower MSEs are obtained when $k_1 = k_2$, while for the asymmetrical settings (scenarios 4, 5 and 6), lower MSEs are obtained for some value of $k_1 > k_2$ (k_1 around 13 in these conditions). This means that better performance is achieved when more transmitted components are allocated to the sources that have more power and are less correlated. We also notice that the modified greedy KLT outperforms the original greedy-algorithm KLT in all the scenarios. The performance of both algorithms though can be in some situations close to or even

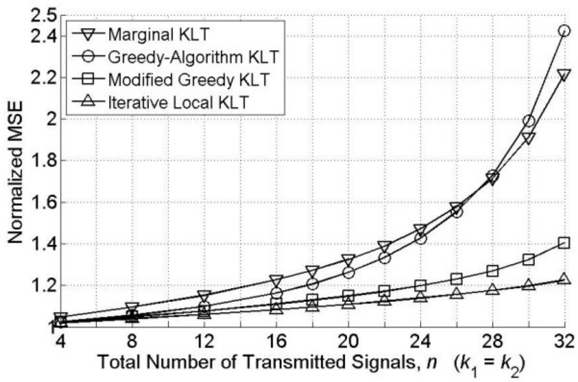

Fig. 7. Normalized MSE performance of different distributed KLT algorithms versus the number of transmitted signals for scenario 1.

worse than the performance of the marginal KLT. In particular, the greedy-algorithm suffers considerably in the highly correlated scenarios 2, 3 and 4. Notice that the expected value of the covariance matrix in (14) for scenario 3 is symmetric Toeplitz with first row $(1, \rho, \cdots, \rho^{N-1})$. References [1] and [4] also use such a matrix in some examples, however in their case the components of vector $\boldsymbol{x}_1$ are interspersed with those of vector $\boldsymbol{x}_2$.

Then the influence of n on the performance of the algorithms was assessed. It is well known that the MSE of the joint KLT decreases as the number of transmitted components increases; the same behaviour can be shown to be true for the other KLT algorithms here studied. To better perceive the differences among the algorithms, we consider the performance of the distributed algorithms relative to the performance of the joint KLT. Fig. 7 compares the various distributed KLT algorithms in scenario 1; the number of transmitted signals in each terminal is the same, and the performance criterion is the MSE normalized to the MSE of the joint KLT. We realize that for large n, the performance of the greedy-algorithm can become worse than that one of the marginal KLT, while the performance of the modified algorithm is kept close to that one of the iterative KLT. The fact that the normalized MSE increases with n for all the algorithms in Fig. 7 might seem counterintuitive, but it only shows that as n grows the MSE of the joint KLT decreases faster than the MSE of the distributed algorithms.

Finally, both greedy algorithms were evaluated as estimators of the optimum size k_i of the encoded vectors, as explained in Section 4.3. Once these values were found, they were used with the iterative local KLT algorithm to obtain the final distributed KLT. At the same time, the iterative local KLT algorithm was run to find the optimum value of k_1 through an exhaustive search. Fig. 8 shows the histograms, averaged over the six scenarios of simulation, of the difference between the optimum value of k_1 and that one estimated by the greedy algorithms. We can see that nearly half of the time both algorithms make a correct estimation of the optimum value of k_1, and nearly the other half they get it wrong by only one unit. From Figs. 3 to 6, we notice that

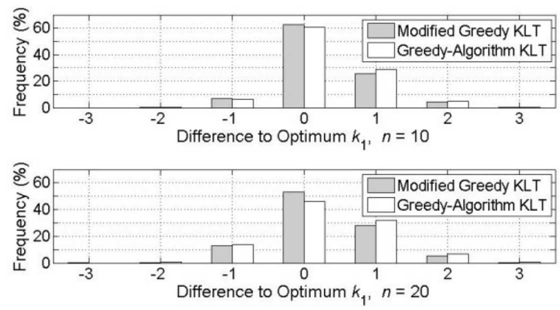

Fig. 8. Evaluation of distributed KLT algorithms as estimators of the optimum size k_i of the encoded vectors.

around the optimum value of k_1 the MSE curves of the iterative local KLT are fairly flat, so a shift of one unit in the value of k_1 might not be as detrimental. In fact, for the MSE normalized to the optimum MSE of the full-search iterative local KLT, we found that for the greedy-algorithm, $MSE = 1.0023$ with $n = 10$ and $MSE = 1.0051$ with $n = 20$, and for the modified algorithm, $MSE = 1.0020$ with $n = 10$ and $MSE = 1.0035$ with $n = 20$.

6. CONCLUSIONS

In this paper, we evaluated the performance of several distributed KLT algorithms based on random covariance matrices generated according to a proposed multi-terminal Markov-chain model. We introduced a modification to the greedy-algorithm KLT that showed to outperform the original algorithm with an increase in complexity. We found though that we could not always rely on any of these algorithms since, in some conditions, their performance can be close to or even worse than that one of the marginal KLT. Nonetheless, simulations showed that the performance of the distributed KLT is highly influenced by the allocation of transmitted signals among the terminals, and both greedy algorithms showed in general an adequate performance for a promising application where the best allocation of the transmitted signals between the terminals is to be obtained.

7. REFERENCES

[1] Gastpar, M., Dragotti, P.L. and Vetterli, M., "The Distributed Karhunen-Loève Transform," *IEEE Trans. on Information Theory*, Vol. 52, No. 12, pp. 5177-5196, Dec. 2006.

[2] Gastpar, M., Vetterli, M. and Dragotti, P.L., "Distributed Signal Processing in Sensor Networks," *IEEE Signal Processing Magazine*, Vol. 23, No. 4, pp. 70-83, July 2006.

[3] Schizas, I.D., Giannakis, G.B., and Luo, Z.Q., "Distributed Estimation Using Reduced-Dimensionality Sensor Observations," *IEEE Trans. on Signal. Proc.*, Vol. 55, pp. 4284-4299, Aug. 2007.

[4] Amar, A., Leshem, A. and Gastpar, M., "Recursive Implementation of the Distributed Karhunen-Loève Transform," *IEEE Trans. on Signal Proc.*, Vol. 58, pp. 5320-5330, Oct. 2010.

[5] Goyal, V.K., "Theoretical Foundations of Transform Coding," *IEEE Signal Proc. Magazine*, Vol. 18, No. 5, pp. 9-21, Sep. 2001.