Signal Processing 104 (2014) 120–135

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed PCA ☆

Alexander Bertrand *,♣, Marc Moonen ♣

KU Leuven, Department of Electrical Engineering ESAT, Stadius Center for Dynamical Systems, Signal Processing and Data Analytics, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

ARTICLE INFO

Article history:
Received 6 July 2013
Received in revised form
6 December 2013
Accepted 26 March 2014
Available online 3 April 2014

Keywords:
Wireless sensor networks
Distributed estimation
Eigenvalue decomposition
Eigenvectors
Principal component analysis

ABSTRACT

We describe a distributed adaptive algorithm to estimate the eigenvectors corresponding to the Q largest or smallest eigenvalues of the network-wide sensor signal covariance matrix in a wireless sensor network. The proposed algorithm recursively updates the eigenvector estimates without explicitly constructing the full covariance matrix that defines them, i.e., without centralizing all the raw sensor signal observations. By only sharing fused Q-dimensional observations, each node obtains estimates of (a) the node-specific entries of the Q covariance matrix eigenvectors, and (b) Q-dimensional projections of the full set of sensor signal observations onto the Q eigenvectors. We also explain how the latter can be used for, e.g., compression and reconstruction of the sensor signal observations based on principal component analysis (PCA), in which each node acts as a data sink. We describe a version of the algorithm for fully-connected networks, as well as for partially-connected networks. In the latter case, we assume that the network has been pruned to a tree topology to avoid cycles in the network. We provide convergence proofs, as well as numerical simulations to demonstrate the convergence and optimality of the algorithm.

1. Introduction

1.1. Context and contribution

The eigenvectors of a signal covariance matrix play an important role in many algorithms and applications, e.g., in principal component analysis (PCA) [2,3], the Karhunen–Loeve transform (KLT) [4], steering vector or direction-of-arrival estimation [5,6], total least squares (TLS) estimation [7], subspace estimation, etc. In this paper, we address the estimation of the eigenvectors of the network-wide sensor signal covariance matrix in a wireless sensor network (WSN). Assume node k collects observations of a node-specific stochastic vector $\mathbf{y}_k$ and let $\mathbf{y}$ be the vector in which all $\mathbf{y}_k$'s are stacked. Our goal is then to adaptively

☆ The work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation – Flanders (FWO). This work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council CoE EF/05/006 'Optimization in Engineering' (OPTEC) and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P7/23 (BESTCOM, 2012-2017) and IUAP P7/19 (DYSCO, 2012-2017), Research Project FWO nr. G.0763.12 'Wireless acoustic sensor networks for extended auditory communication' and project HANDiCAMS. The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944. The scientific responsibility is assumed by its authors. A conference precursor of this manuscript has been published as [1].

* Corresponding author. Tel.: +32 16 321899; fax: +32 16 321970.
E-mail addresses: alexander.bertrand@esat.kuleuven.be (A. Bertrand), marc.moonen@esat.kuleuven.be (M. Moonen).
♣ EURASIP member.

http://dx.doi.org/10.1016/j.sigpro.2014.03.037
0165-1684/

estimate the Q eigenvectors corresponding to the Q largest or smallest eigenvalues of the network-wide covariance matrix defined by $\mathbf{y}$. In principle, this would require each node to transmit its raw sensor signal observations to a central node or fusion center (FC), where the network-wide covariance matrix can be constructed, after which an eigenvalue decomposition (EVD) can be performed. However, centralizing all these raw observations may require too much communication bandwidth, in particular if observations are collected at a high sampling rate, as in audio or video applications. Furthermore, if $\mathbf{y}$ has a large dimension, the computation of the EVD of the network-wide covariance matrix may require a significant amount of computational power at the FC since the computational complexity of the EVD scales cubically with the matrix dimension.

To reduce the communication and computation cost, we propose a distributed adaptive algorithm to estimate Q eigenvectors without explicitly constructing the network-wide covariance matrix that actually defines them, i.e., without the need to centralize the sensor signal observations in an FC. Instead of transmitting all its raw sensor signal observations, each node only transmits fused/compressed Q-dimensional observations, while estimating the node-specific entries of the eigenvector, corresponding to the part of $\mathbf{y}$ that is observed at the node. We refer to the algorithm as the distribute adaptive covariance matrix eigenvector estimation (DACMEE) algorithm.

The DACMEE algorithm also provides each node with the Q-dimensional projections of the full set of sensor signal observations onto the Q estimated eigenvectors. This allows each node to compute a PCA- or KLT-based approximation of the observations of the full network-wide vector $\mathbf{y}$.

We will describe two versions of the DACMEE algorithm, i.e., a version for fully-connected networks in which a signal broadcast by any node can be collected by every other node, as well as a version for partially-connected networks in which a node can only communicate with a subset of the other nodes. We then assume that the partially-connected network is pruned to a tree topology. This guarantees that there are no cycles in the network graph, since these harm the algorithm dynamics.

1.2. Relation to prior work

Two different cases have been considered in the literature where either (a) the nodes collect observations of the full vector $\mathbf{y}$, or (b) each node collects observations of a node-specific subset of the entries of $\mathbf{y}$ (as it is the case in this paper). Let $\mathbf{Y}$ denote an $M \times N$ observation matrix containing N observations of an M-dimensional stochastic vector $\mathbf{y}$, then (a) corresponds to the case where the *columns* of $\mathbf{Y}$ are distributed over the different nodes, whereas in case (b), the *rows* of $\mathbf{Y}$ are distributed over the nodes. The techniques to construct the corresponding covariance matrix and/or estimate its eigenvectors are very different for the two cases. It is noted that a similar distinction exists in the literature in the context of distributed least-squares estimation, see, e.g., [8,9], where each node collects observations of the full $\mathbf{y}$ to estimate a common parameter vector, versus [10,11], where each node only observes a node-specific subset of the entries of $\mathbf{y}$ to compute an estimator that relies on the full network-wide covariance matrix.

Case (a) is addressed in [12–14] for ad hoc topologies and in [15] for a fully-connected topology. In [12], the network-wide covariance matrix is first estimated by means of a consensus averaging (CA) algorithm that exchanges $M \times M$ matrices in each iteration, after which each node performs a local EVD. If only a subset of the eigenvectors[1] is needed, one can use distributed optimization techniques in which only M-dimensional vectors are exchanged between nodes [13,14]. In [15], a distributed QR decomposition is performed in a fully-connected network, followed by an EVD.

Case (b) is actually more challenging, as it requires to estimate the cross-correlation between sensor signals of different nodes. This requires the exchange of (compressed) sensor signal observations, resulting in a higher communication cost compared to case (a), in particular for applications with a high sampling rate. Case (b) is tackled in [6,16,17] (only for the case of principal eigenvectors) for networks with an ad hoc topology. These algorithms rely on Oja's learning rule in combination with nested CA iterations, hence operating at two time scales. The inner loop performs many CA iterations with a full reset for each outer loop iteration. Since the outer loop runs with the same rate as the sampling rate of $\mathbf{y}$, and since each iteration of the inner loop also requires data exchange, each node actually transmits more data than actually collected by its sensors. Furthermore, since the convergence time of the inner CA loop increases with the network size, the per-node communication cost also grows with the network size.

The algorithm proposed in this paper only works in networks with a fully connected or a tree topology, but it does not require nested loops, and its per-node communication cost is independent of the network size. The algorithm does not explicitly rely on Oja's stochastic learning rule (although this can also be included), but it explicitly computes compressed sensor signal covariance matrices at each node. The latter allows us to, e.g., remove the effect of spatially correlated noise by subtracting a known or estimated noise covariance matrix from the local covariance matrices. The algorithm is also able to estimate the eigenvectors corresponding to the smallest eigenvalues (e.g., for TLS estimation), which is not possible in [6,16,17].

Finally, it is noted that there exists other related work in the context of (b) (see, e.g., [3,4]), which however requires prior knowledge of the network-wide covariance matrix. In our case, the network-wide covariance matrix is assumed to be unknown (and possibly even time-varying).

1.3. Paper outline

The outline of the paper is as follows. Section 2 gives the problem statement as well as an application example

[1] The algorithms in [13,14] estimate the eigenvector corresponding to the smallest eigenvalue of the covariance matrix, but the algorithms are easily adapted to compute the principal eigenvectors.