Decomposable Principal Component Analysis

Ami Wiesel, *Member, IEEE*, and Alfred O. Hero, *Fellow, IEEE*

***Abstract*—In this paper, we consider principal component analysis (PCA) in decomposable Gaussian graphical models. We exploit the prior information in these models in order to distribute PCA computation. For this purpose, we reformulate the PCA problem in the sparse inverse covariance (concentration) domain and address the global eigenvalue problem by solving a sequence of local eigenvalue problems in each of the cliques of the decomposable graph. We illustrate our methodology in the context of decentralized anomaly detection in the Abilene backbone network. Based on the topology of the network, we propose an approximate statistical graphical model and distribute the computation of PCA.**

***Index Terms*—Anomaly detection, graphical models, principal component analysis.**

I. Introduction

We consider principal component analysis (PCA) in Gaussian graphical models. PCA is a classical nonparametric dimensionality reduction method which is frequently used in statistics and machine learning in order to reduce sample variance with minimal loss of information [1], [11]. The first few principal components can be interpreted as the best low-dimensional linear approximation to the sample. On the other hand, Gaussian graphical models, also known as covariance selection models, exploit conditional independence structure within the assumed multivariate Gaussian sample distribution [7], [16]. These models represent the sample distribution on a graph, and allow for efficient distributed implementation of statistical inference algorithms, e.g., the well-known belief propagation method and the junction tree algorithm [13], [20]. In particular, decomposable graphs, also known as chordal or triangulated graphs, provide computationally simple inference methods. Our main contribution is the application of decomposable graphical models to PCA which we nickname DPCA, where "D" denotes both *Decomposable* and *Distributed*.

The main motivation for distributed PCA is decentralized dimensionality reduction. It plays a leading role in distributed estimation and compression theory in wireless sensor networks [9], [18], [19], [21], [23], and decentralized data mining techniques [2], [14], [17]. It is also used in anomaly detection in computer networks [6], [12], [15]. In particular, [9] and [18] proposed to approximate the global PCA using a sequence of conditional local PCA solutions. Alternatively, an approximate solution which allows a tradeoff between performance and communication requirements was proposed in [12] using eigenvalue perturbation theory.

DPCA is an efficient implementation of distributed PCA based on a prior graphical model. Unlike the above references it does not try to approximate PCA, but yields an exact solution up to on any given error tolerance. DPCA assumes additional prior knowledge in the form of a graphical model that is not taken into account by previous distributed PCA methods. Such models are very common in signal processing and communications. For example, the Gauss Markov source example in [9] and [18] is probably the most celebrated decomposable graphical model. In some problems approximate conditional independency structure can be learned from the observed data using methods such as [3], [8], and [22]. Alternatively, conditional independence can sometimes be surmised from other nonstatistical prior knowledge. For example, the known topology of a sensor network can be used to infer that, given data at neighboring sensors, the data at a given sensor is conditionally independent of data at distant sensors [5]. When one cannot identify any obvious conditional independence structure, DPCA can be interpreted as an approximate PCA method that allow one to trade accuracy for decentralized scalability by introducing sparsity. If the model is not decomposable then an approximation can be obtained using classical graph theoretic algorithms [13]. In any case, once the conditional dependency relationships specify a decomposable model our DPCA algorithm will be applicable.

PCA can be interpreted as maximum likelihood (ML) estimation of the covariance followed by its eigenvalue decomposition (EVD). When the samples follow a Gaussian graphical model, PCA can be performed by applying the EVD to the ML estimator of the model parameters. However, this approach to PCA does not exploit the structure of the graphical model, whereas DPCA is specifically designed to fully exploit it. DPCA is formulated in the sparse concentration (inverse covariance) domain in which the global EVD is successively approximated by a sequence of local EVD's and a small amount of message passing. The local EVD's are solved over each clique in the decomposable graph associated with the graphical model. When the DPCA algorithm terminates, each clique obtains its own local version of the principal components.

To illustrate DPCA we apply it to distributed anomaly detection in computer networks [12], [15]. In this context, DPCA learns a low dimensional graphical model of the normal traffic behavior and performs outlier detection. This application is nat-

Manuscript received August 19, 2008; revised May 04, 2009. First published June 19, 2009; current version published October 14, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. C. Guillemot. The work of A. Wiesel was supported by a Marie Curie Outgoing International Fellowship within the 7th European Community Framework Programme. This work was also partially supported by Air Force Office of Scientific Research under Grant FA9550-06-1-0324.

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: amiw@umich.edu; hero@umich.edu).

Digital Object Identifier 10.1109/TSP.2009.2025806

1053-587X/$26.00 © 2009 IEEE

ural since the network's physical topology provides a justification for an approximate graphical model. For example, consider two nodes which are geographically distant and linked only through a single long path of nodes. It is reasonable to model these two sensors as independent conditioned on the path. We examine the validity of this model in the context of anomaly detection in the Abilene network using real-world Internet traffic. We propose an approximate decomposition of the Abilene network, enable the use of DPCA and obtain a fully distributed anomaly detection method.

The outline of the paper is as follows. Decomposable graphs are most easily understood for the case of two cliques. Therefore, we begin in Section II by introducing the problem formulation and solution to DPCA in this simple case. The generalization to arbitrary decomposable graphs is presented in Section III, which depends on a recursive application of the two clique solution. Then in Sections IV and V, we demonstrate the use of DPCA using two examples. First, in Section IV, we simulate our proposed algorithm in a synthetic tracking scenario. Second, in Section V, we illustrate its application to anomaly detection using a real-world dataset from the Abilene backbone network. Finally, in Section VI, we provide concluding remarks and address future work.

The following notation is used. Boldface upper case letters denote matrices, boldface lower case letters denote column vectors, and standard lower case letters denote scalars. The superscripts $(\cdot)^T$ and $(\cdot)^{-1}$ denote the matrix transpose and matrix inverse, respectively, $|a|$ is the cardinality of a, $a \cup b$ is the union of the sets a and b, and $a \setminus b$ is the set of elements in a which are not in b. The matrix $\mathbf{I}$ denotes the identity, $\mathrm{eig}_{\min}(\mathbf{X})$ is the minimum eigenvalue of square symmetric matrix $\mathbf{X}$, $\mathbf{u}_{\mathrm{null}}(\mathbf{X})$ is a null vector of $\mathbf{X}$, $\mathrm{eig}_{\max}(\mathbf{X})$ is the maximum eigenvalue of $\mathbf{X}$, and $\mathbf{X} \succ \mathbf{0}$ means that $\mathbf{X}$ is positive definite. Finally, we use indices in the subscript $[\mathbf{x}]_a$ or $[\mathbf{X}]_{a,b}$ to denote subvectors or submatrices, respectively, and $[\mathbf{X}]_{a,:}$ denotes the submatrix formed by the a'th rows in $\mathbf{X}$. Where possible, we omit the brackets and use $\mathbf{x}_a$ or $\mathbf{X}_{a,b}$ instead.

II. Two Clique DPCA

In this section, we introduce DPCA for a simple case which will become the building block for the general algorithm.

A. Problem Formulation

Let $\mathbf{x}$ be a length p, zero mean Gaussian random vector of covariance $\mathbf{\Sigma}$. We partition the vector $\mathbf{x} = [\mathbf{x}_a^T \quad \mathbf{x}_c^T \quad \mathbf{x}_b^T]^T$ where a, c and b are disjoint subsets of indices. For later use[1], we define two cliques of indices $C_1 = \{a, c\}$ and $C_2 = \{c, b\}$, as well as the history subset $H_1 = \{a, c\}$, the separator subset $S_2 = \{c\}$ and the remainder subset $R_2 = \{b\}$.

The key assumption underlying DPCA is that $[\mathbf{x}]_a$ and $[\mathbf{x}]_b$ are independent conditioned on $[\mathbf{x}]_c$

$$[\mathbf{x}]_a \perp [\mathbf{x}]_b \,|\, [\mathbf{x}]_c. \tag{1}$$

Due to the properties of the conditional Gaussian distribution, this implies that

$$[\mathbf{\Sigma}]_{a,b} - [\mathbf{\Sigma}]_{a,c}\left([\mathbf{\Sigma}]_{c,c}\right)^{-1}[\mathbf{\Sigma}]_{c,b} = \mathbf{0}. \tag{2}$$

Using the inversion formula for block partitioned matrices, (2) results in

$$\left[\mathbf{\Sigma}^{-1}\right]_{a,b} = \left[\mathbf{\Sigma}^{-1}\right]_{b,a}^T = \mathbf{0}. \tag{3}$$

The input to DPCA is a set of n independent and identically distributed realizations of $\mathbf{x}$, denoted by $\mathbf{x}_i$ for $i = 1, \ldots, n$. More specifically, this input is distributed in the sense that the first clique only has access to $[\mathbf{x}_i]_{C_1}$ for $i = 1, \ldots, n$, whereas the second clique only has access to $[\mathbf{x}_i]_{C_2}$ for $i = 1, \ldots, n$. The covariance $\mathbf{\Sigma}$ is unknown, but we assume prior knowledge of the conditional independence structure defined by (3). Using local data and message passing between the two cliques, DPCA searches for the linear combination $X = \mathbf{u}^T\mathbf{x}$ having maximal variance. When the algorithm terminates, each of the cliques obtains its own local version of $\mathbf{u}$, i.e., the subvectors $[\mathbf{u}]_{C_1}$ and $[\mathbf{u}]_{C_2}$.

The following subsections present the proposed solution to DPCA. It involves two main stages: covariance estimation and principal components computation.

B. Solution: Covariance Matrix Estimation

First, the covariance matrix of $\mathbf{x}$ is estimated using the maximum likelihood (ML) technique. Specifically, the ML estimate of $\mathbf{\Sigma}$, denoted by $\mathbf{C}$, is defined as the parameter which maximizes the likelihood of the observations subject to the constraint (3). Due to the special decomposable structure, the solution is simply [16, Prop. 5.6, p. 138]

$$\mathbf{C} = \mathbf{K}^{-1} \tag{4}$$

where $\mathbf{K}$ is

$$\mathbf{K} = \begin{bmatrix} \left([\mathbf{S}]_{C_1,C_1}\right)^{-1} & \mathbf{0} \\ & \mathbf{0} \\ \mathbf{0} \quad \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \quad \mathbf{0} \\ \mathbf{0} & \left([\mathbf{S}]_{C_2,C_2}\right)^{-1} \\ \mathbf{0} & \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \left([\mathbf{S}]_{S_2,S_2}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5}$$

and $\mathbf{S}$ is the sample covariance

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T. \tag{6}$$

Thus, the ML estimate $\mathbf{C}$ is a dense matrix, but its inverse $\mathbf{K}$ (which is actually the ML estimate of the concentration matrix) has an appealing sparse structure. Moreover, $\mathbf{K}$ can be easily computed in a distributed manner. Each clique can compute its

[1] This terminology will be made precise in Section III when we extend the results to decomposable graphical models.

subblock of $\mathbf{K}$ using only local information and a single message from the other clique of dimension $|S_2|$, which may be small

$$[\mathbf{K}]_{C_1,C_1} = \left([\mathbf{S}]_{C_1,C_1}\right)^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{2\rightarrow 1} \end{bmatrix} \qquad (7)$$

$$[\mathbf{K}]_{C_2,C_2} = \left([\mathbf{S}]_{C_2,C_2}\right)^{-1} + \begin{bmatrix} \mathbf{M}_{1\rightarrow 2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad (8)$$

where the messages $\mathbf{M}_{2\rightarrow 1}$ and $\mathbf{M}_{1\rightarrow 2}$ are defined to satisfy (5). The other elements in $\mathbf{K}$ which do not belong to either clique are equal to zero

$$[\mathbf{K}]_{ab} = [\mathbf{K}]_{ba}^T = \mathbf{0}. \qquad (9)$$

C. Solution: First Principal Eigenvalue

Given the ML covariance estimate $\mathbf{C}$, the PCA objective function is estimated as

$$\mathbf{u}^T\mathbf{C}\mathbf{u} \qquad (10)$$

and maximized subject to a norm constraint to yield

$$\mathrm{eig}_{\max}(\mathbf{C}) = \begin{cases} \max_{\mathbf{u}}, & \mathbf{u}^T\mathbf{C}\mathbf{u} \\ \text{s.t.}, & \mathbf{u}^T\mathbf{u} = 1. \end{cases} \qquad (11)$$

This optimization gives both the maximal eigenvalue of $\mathbf{C}$ and its eigenvector $\mathbf{u}$.

The drawback to the above solution is that the EVD computation requires centralized processing and does not exploit the structure of $\mathbf{K}$. Each clique needs to send its local covariance to a central processing unit which constructs $\mathbf{C}$ and computes its maximal eigenvalue and eigenvector. We will now provide an alternative distributed DPCA algorithm in which each clique uses only local information along with message passing in order to calculate its local version of $\mathrm{eig}_{\max}(\mathbf{C})$ and $\mathbf{u}$.

Our first observation is that DPCA can be equivalently solved in the concentration domain instead of the covariance domain. Indeed, it is well known that

$$\mathrm{eig}_{\max}(\mathbf{C}) = \frac{1}{\mathrm{eig}_{\min}(\mathbf{K})} \qquad (12)$$

when the inverse exists. The corresponding eigenvectors are also identical. The advantage of working with $\mathbf{K}$ instead of $\mathbf{C}$ is that we can exploit $\mathbf{K}$'s sparsity as expressed in (9).

Before continuing it is important to address the question of singularity of $\mathbf{C}$. One may claim that working in the concentration domain is problematic since $\mathbf{C}$ may be singular. This is indeed true but is not a critical disadvantage since graphical models allow for well conditioned estimates under small sample sizes. For example, classical ML exists only if $n \geq p$, whereas the ML described above requires the less stringent condition $n \geq \max\{|C_1|, |C_2|\}$ [16, Prop. 5.6, p. 138].

We now return to the problem of finding

$$\lambda = \mathrm{eig}_{\min}(\mathbf{K}) \qquad (13)$$

in a distributed manner. We begin by expressing λ as a trivial line-search problem

$$\lambda = \sup \quad t \quad \text{s.t.} \quad t < \mathrm{eig}_{\min}(\mathbf{K}) \qquad (14)$$

and note that the objective is linear and the constraint set is convex. It can be solved using any standard line-search algorithm, e.g., bisection. At first, this representation seems useless as we still need to evaluate $\mathrm{eig}_{\min}(\mathbf{K})$ which was our original goal. However, the following proposition shows that checking the feasibility of a given t can be done in a distributed manner.

Proposition 1: Let $\mathbf{K}$ be a symmetric matrix with structure

$$\mathbf{K} = \begin{bmatrix} \multicolumn{2}{c}{\mathbf{K}_{H_1,H_1}} & \mathbf{0} \\ & & \mathbf{K}_{S_2,R_2} \\ \mathbf{0} & \mathbf{K}_{R_2,S_2} & \mathbf{K}_{R_2,R_2} \end{bmatrix}. \qquad (15)$$

Then, the constraint

$$t < \mathrm{eig}_{\min}(\mathbf{K}) \qquad (16)$$

is equivalent to the following pair of constraints:

$$t < \mathrm{eig}_{\min}(\mathbf{K}_{R_2,R_2}) \qquad (17)$$

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}_{H_1,H_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}(t) \end{bmatrix}\right) \qquad (18)$$

with the *message matrix* defined as

$$\mathbf{M}(t) = \mathbf{K}_{S_2,R_2}\left(\mathbf{K}_{R_2,R_2} - t\mathbf{I}\right)^{-1}\mathbf{K}_{R_2,S_2}. \qquad (19)$$

Proof: The proof is obtained by rewriting (16) as a linear matrix inequality

$$\mathbf{K} - t\mathbf{I} \succ \mathbf{0} \qquad (20)$$

and decoupling (20) using the following lemma.

Lemma 1 (Schur's Lemma [4, Appendix A5.5]): Let $\mathbf{X}$ be a symmetric matrix partitioned as

$$\mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}. \qquad (21)$$

Then, $\mathbf{X} \succ \mathbf{0}$ if and only if $\mathbf{C} \succ \mathbf{0}$ and $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \succ \mathbf{0}$.

Applying Schur's Lemma to (20) with $\mathbf{C} = \mathbf{K}_{R_2,R_2} - t\mathbf{I}$ and rearranging yields

$$t\mathbf{I} \prec \mathbf{K}_{R_2,R_2} \qquad (22)$$

$$t\mathbf{I} \prec \mathbf{K}_{H_1,H_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}(t) \end{bmatrix}. \qquad (23)$$

Finally, (17) and (18) are obtained by rewriting (22) and (23) as eigenvalue inequalities, respectively. ■

Proposition 1 provides an intuitive distributed solution to (14). For any given t we can check the feasibility by solving local eigenvalue problems and message passing via $\mathbf{M}(t)$ whose dimension is equal to the cardinality of the separator. The optimal global eigenvalue is then defined as the maximal globally feasible t. Specifically, in Algorithm 1 (displayed below) we provide a pseudo code for the two clique DPCA that solves for t using the bisection method. Given initial bounds

$$0 \leq \mathrm{eig}_{\min}\left(\mathbf{K}\right) \leq U \tag{24}$$

Algorithm 1 is guaranteed to find the minimal eigenvalue up to any required tolerance ϵ within $\log_2(U - L/\epsilon)$ iterations. Each iteration consists of up to two local eigenvalue problems for testing (17)–(18) and transmission of a single message from C_2 to C_1 of size $|S_2| \times |S_2|$. A simple choice for the bounds in (24) is $L = 0$ since $\mathbf{K}$ is positive definite, and

$$U = \min\left\{\mathrm{eig}_{\min}\left(\mathbf{K}_{H_1,H_1}\right), \mathrm{eig}_{\min}\left(\mathbf{K}_{R_2,R_2}\right)\right\} \tag{25}$$

as proved in the Lemma 2 in the Appendix.

Algorithm 1: Bisection line search for two cliques DPCA

```
Input: $\mathbf{K}$, $L$, $U$, $\epsilon$, $H_1$, $S_2$, $R_2$
Output: $t$
while $U - L > \epsilon$ do
    $t = (U + L)/2$
    $\mathbf{Q} = \mathbf{K}$
    if $t < \mathrm{eig}_{\min}(\mathbf{Q}_{R_2,R_2})$ then
        $\mathbf{M}(t) = \mathbf{Q}_{S_2,R_2}(\mathbf{Q}_{R_2,R_2} - t\mathbf{I})^{-1}\mathbf{Q}_{R_2,S_2}$
        $\mathbf{Q}_{S_2,S_2} = \mathbf{Q}_{S_2,S_2} - \mathbf{M}(t)$
    else
        $U = t$
    end
    if $t < \mathrm{eig}_{\min}(\mathbf{Q}_{H_1,H_1})$ then
        $L = t$
    else
        $U = t$
    end
end
```

D. Solution: First Principal Eigenvector

After we obtain the minimal eigenvalue λ, we can easily recover its corresponding eigenvector $\mathbf{u}$. For this purpose, we define $\mathbf{Q} = \mathbf{K} - \lambda\mathbf{I}$ and obtain $\mathbf{u} = \mathbf{u}_{\mathrm{null}}\left(\mathbf{Q}\right)$. The matrix $\mathbf{Q}$ follows the same block sparse structure as $\mathbf{K}$, and the linear set of equations $\mathbf{Qu} = \mathbf{0}$ can be solved in a distributed manner. There are two possible solutions. Usually, $\mathbf{Q}_{R_2,R_2}$ is nonsingular in which case it is easy to verify that the solution is

$$[\mathbf{u}]_{H_1} = \mathbf{u}_{\mathrm{null}}\left(\mathbf{Q}_{H_1,H_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}\right) \tag{26}$$

$$[\mathbf{u}]_{R_2} = -\mathbf{Q}_{R_2,R_2}^{-1}\mathbf{Q}_{R_2,S_2}\left[\mathbf{u}\right]_{S_2} \tag{27}$$

where the *message* $\mathbf{M}$ is defined as

$$\mathbf{M} = \mathbf{Q}_{S_2,R_2}\mathbf{Q}_{R_2,R_2}^{-1}\mathbf{Q}_{R_2,S_2}. \tag{28}$$

Otherwise, if $\mathbf{Q}_{R_2,R_2}$ is singular then the solution is

$$[\mathbf{u}]_{H_1} = \mathbf{0} \tag{29}$$

$$[\mathbf{u}]_{R_2} = \mathbf{u}_{\mathrm{null}}\left(\mathbf{Q}_{R_2,R_2}\right) \tag{30}$$

since

$$[\mathbf{Qu}]_{H_1} = [\mathbf{Q}]_{H_1,H_1}\mathbf{0} + [\mathbf{Q}]_{H_1,R_2}\mathbf{u}_{\mathrm{null}}\left(\mathbf{Q}_{R_2,R_2}\right) = \mathbf{0}. \tag{31}$$

due to Lemma 3 in the appendix. The singular case is highly unlikely as the probability of (29) in continuous models is zero. However, for finite register length computations this condition needs to be checked.

E. Solution: Higher Order Components

In practice, dimensionality reduction involves the projection of the data into the subspace of a few principal components. A standard approach for computing higher order components is the deflation method, i.e., iteratively solving for the first order component of deflated (modified) matrices. We now apply this approach to DPCA.

The j'th order principal component is defined as the linear transformation which is orthogonal to the preceding components and preserves maximal variance. It is given by $\mathbf{X}_j = \mathbf{u}_j^T\mathbf{x}$ where $\mathbf{u}_j$ is the j'th principal eigenvector of $\mathbf{C}$. In the concentration domain, $\mathbf{u}_j$ is the eigenvector associated with λ_j, the j'th smallest eigenvalue of $\mathbf{K}$. These high order eigenvalues and eigenvectors can be found by iteratively applying the previous algorithms to deflated versions of $\mathbf{K}$. Specifically, let

$$\lambda_1 \leq \cdots \leq \lambda_p \tag{32}$$

be the ordered eigenvalues of $\mathbf{K}$ with corresponding eigenvalues $\mathbf{u}_1, \ldots, \mathbf{u}_p$. We define the deflated matrix as

$$\overline{\mathbf{K}} = \mathbf{K} + \mathbf{UDU}^T \tag{33}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}]$, $\mathbf{D} = d\mathbf{I}$ and d is a sufficiently large positive constant. It is easy to verify that $\overline{\mathbf{K}}$ and $\mathbf{K}$ have the same eigenvectors, and that the unordered eigenvalues of $\overline{\mathbf{K}}$ are

$$\lambda_1 + d, \ldots, \lambda_{j-1} + d, \lambda_j, \ldots, \lambda_p. \tag{34}$$

Assuming that d is sufficiently large, the jth smallest eigenvalue of $\mathbf{K}$ is equal to the minimal eigenvalue of $\overline{\mathbf{K}}$ and can be found as the solution to

$$\lambda_j = \sup \quad t \quad \text{s.t.} \quad t < \mathrm{eig}_{\min}\left(\overline{\mathbf{K}}\right). \tag{35}$$

The matrix $\overline{\mathbf{K}}$ does not necessarily satisfy the sparse block structure of $\mathbf{K}$, and we cannot use Proposition 1 directly. Fortunately, this proposition can be easily modified to address low rank perturbations.

Proposition 2: Let $\overline{\mathbf{K}}$ be a symmetric matrix with structure

$$\overline{\mathbf{K}} = \begin{bmatrix} \multicolumn{2}{c}{\mathbf{K}_{H_1,H_1}} & \mathbf{0} \\ & & \mathbf{K}_{S_2,R_2} \\ \mathbf{0} & \mathbf{K}_{R_2,S_2} & \mathbf{K}_{R_2,R_2} \end{bmatrix} + \mathbf{UDU}^T. \tag{36}$$

Then, the constraint

$$t < \mathrm{eig}_{\min}\left(\overline{\mathbf{K}}\right) \tag{37}$$

is equivalent to the following pair of constraints

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}_{R_2,R_2} + [\mathbf{U}]_{R_2,:}\mathbf{D}\left[\mathbf{U}\right]_{R_2,:}^T\right) \tag{38}$$

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}_{H_1,H_1} + \left[\overline{\mathbf{U}}\right]_{H_1,:}\overline{\mathbf{D}}\left[\overline{\mathbf{U}}\right]_{H_1,:}^T\right) \tag{39}$$

where

$$[\overline{\mathbf{U}}]_{H_1,:} = \left[\begin{matrix}\mathbf{0}\\ \mathbf{I}\end{matrix} \quad [\mathbf{U}]_{H_1}\right] \tag{40}$$

$$\overline{\mathbf{D}} = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\end{bmatrix} - \mathbf{M}_{\mathbf{U}}(t) \tag{41}$$

and the *message matrix* $\mathbf{M}_{\mathbf{U}}(t)$ is defined as

$$\mathbf{M}_{\mathbf{U}}(t) = \begin{bmatrix}\mathbf{K}_{S_2,R_2}\\ \mathbf{D}[\mathbf{U}]^T_{R_2,:}\end{bmatrix} \times \left(\mathbf{K}_{R_2,R_2} + [\mathbf{U}]_{R_2,:}\mathbf{D}[\mathbf{U}]^T_{R_2,:} - t\mathbf{I}\right)^{-1} \begin{bmatrix}\mathbf{K}_{S_2,R_2}\\ \mathbf{D}[\mathbf{U}]^T_{R_2,:}\end{bmatrix}^T. \tag{42}$$

Proof: The proof is similar to that of Proposition 1 and therefore omitted. ■

Thus, the method in Section II-C can be adjusted to find the j'th smallest eigenvalue. The only difference is that the messages are slightly larger than before. Each message is a matrix of size $(|S_2|+j-1)\times(|S_2|+j-1)$ instead of $|S_2|\times|S_2|$.

F. Solution: Communication and Computation Complexity

The main advantage in DPCA is its distributed implementation and reduced communication requirements. Classical PCA requires each node to send all of its samples to a centralized processing unit which receives these np samples and computes the global sample covariance. Once this matrix is computed there is no need for further communication. On the other hand, DPCA partitions the nodes into overlapping cliques. Each clique has a local processing terminal that collects its data (a total of $n|C_i|$ samples where $i = 1, 2$) and computes its local sample covariance. Next, each eigenvalue computation requires additional message passing of $I(|S|+j-1)^2$ variables where I is the number of iterations and j is the eigenvalue order. Thus, DPCA is advantageous when a few conditions are met. First, the cost of local communication within the clique should be negligible in comparison to global communication. The second condition is that the number of coupling nodes $|S|$ and the number of required principal components j are sufficiently small so that $(|S|+j-1)^2 \ll p$.

III. DPCA in Decomposable Graphs

We now proceed to the general problem of DPCA in decomposable graphs. In the previous section, we showed that DPCA can be computed in a distributed manner if it is a priori known that $\mathbf{x}_a$ and $\mathbf{x}_b$ are conditionally independent given $\mathbf{x}_c$. Graphical models are intuitive characterizations of such conditional independence structures. In particular, decomposable models are special graphs that can be recursively subdivided into the two cliques graph in Fig. 1. This section gives the formal definitions of decomposable graphs [16], followed by a recursive application of the previous two-clique DPCA algorithm.

An undirected graph $\mathcal{G} = (V, E)$ is a set of nodes V connected by undirected edges E. A graph (or subgraph) is complete if all of its nodes are connected by an edge. A subset $c \subseteq V$ separates $a \subseteq V$ and $b \subseteq V$ if all paths from a to b intersect c. A triple $\{a, c, b\}$ of disjoint subsets of $\mathcal{G}$ is a decomposition of $\mathcal{G}$ if $V = a \cup c \cup b$, c separates a and b, and c is complete. Finally, a graph $\mathcal{G}$ is decomposable if it is complete, or if it can be recursively decomposed into two decomposable subgraphs $\mathcal{G}_{a\cup c}$ and $\mathcal{G}_{c\cup b}$.

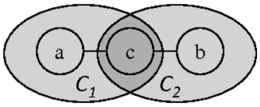

Fig. 1. Graphical model with two cliques modeling a 3 node network in which a and b are conditionally independent given c.

It is convenient to define a decomposable graph using cliques. A clique is a maximal complete subset of nodes. Any decomposable graph can be represented using a sequence of cliques $C_1, \ldots, C_K$ which satisfy a *perfect elimination order*. An important property of this order is that S_j separates $H_{j-1}\backslash S_j$ from R_j where

$$H_j = C_1 \cup C_2 \cup \cdots \cup C_j, \quad j = 1, \ldots, K \tag{43}$$

$$S_j = H_{j-1} \cap C_j, \quad j = 2, \ldots, K \tag{44}$$

$$R_j = H_j \backslash H_{j-1}, \quad j = 2, \ldots, K. \tag{45}$$

For example, the two cliques graph in Fig. 1 is a simple decomposable graph with $C_1 = \{a, c\}$, $C_2 = \{c, b\}$, $H_1 = \{a, c\}$, $H_2 = \{a, c, b\}$, $S_2 = \{c\}$ and $R_2 = \{b\}$. Accordingly, $S_2 = \{c\}$ separates $H_1\backslash S_2 = \{a\}$ from $R_2 = \{b\}$.

Based on these definitions, we can now consider graphical models. A random vector $\mathbf{x}$ satisfies the Markov property with respect to $\mathcal{G}$, if for any pair of nonadjacent nodes the corresponding pair of random variables are conditionally independent of the rest of the elements in $\mathbf{x}$. For the Gaussian distribution, this definition results in sparsity of the concentration matrix $\mathbf{K}$, i.e., $[\mathbf{K}]_{i,j} = 0$ for any pair $\{i, j\}$ of nonadjacent nodes. When $\mathcal{G}$ is decomposable, $\mathbf{K}$ has an appealing block sparsity pattern.

Similarly to Section II-B, global ML estimation of the concentration matrix in a decomposable Gaussian graphical model has a simple closed form, which can be computed in a distributed manner [16, Prop. 5.9, p. 146]

$$\mathbf{K} = \sum_{k=1}^{K}\left[\left([\mathbf{S}]_{C_k,C_k}\right)^{-1}\right]^0 - \sum_{k=2}^{K}\left[\left([\mathbf{S}]_{S_k,S_k}\right)^{-1}\right]^0 \tag{46}$$

where $\mathbf{S}$ is the sample covariance defined in (6) and the zero fill-in operator $[\cdot]^0$ outputs a matrix of the same dimension as $\mathbf{K}$ where the argument occupies the appropriate subblock and the rest of the matrix has zero valued elements. (See (5) for a two clique example, and [16] for the exact definition of this operator.)

The eigenvalue computation can also be implemented in a distributed manner by recursively applying Proposition 1. Indeed, Proposition 1 shows that the eigenvalue inequality

$$t < \mathrm{eig}_{\min}(\mathbf{K}) \tag{47}$$

is equivalent to two adjusted local eigenvalue inequalities

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}'_{R_K}(t)\right) \tag{48}$$

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}'_{H_{K-1}}(t)\right) \tag{49}$$

where

$$\mathbf{K}'_{R_K}(t) = \mathbf{K}_{R_K,R_K} \tag{50}$$

$$\mathbf{K}'_{H_{K-1}}(t) = \mathbf{K}_{H_{K-1},H_{K-1}}(t) - [\mathbf{M}_k(t)]^0. \tag{51}$$

and $\mathbf{M}_k(t)$ is a message as in (19). Next, we can apply Schur's Lemma again and replace (49) with two additional inequalities:

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}'_{R_K}(t)\right) \tag{52}$$

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}''_{R_{K-1}}(t)\right) \tag{53}$$

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}''_{H_{K-2}}(t)\right) \tag{54}$$

where $\mathbf{K}''_{R_{K-1}}(t)$ and $\mathbf{K}''_{H_{K-2}}(t)$ are similarly defined. We continue in an iterative fashion until we obtain K decoupled eigenvalue inequalities. Thus, the feasibility of a given t can be checked in a distributed manner via message passing between the cliques.

In Algorithm 2 displayed below we provide a pseudo code for DPCA which solves for t using the bisection method. In the code, we keep $\mathbf{K}$ constant and use the variable $\mathbf{Q}$ to denote its modified versions which were previously labeled using tag superscripts. Given initial bounds

$$L \leq \mathrm{eig}_{\min}(\mathbf{K}) \leq U \tag{55}$$

the algorithm is guaranteed to find the minimal eigenvalue up to any required tolerance ϵ within $\log_2(U - L/\epsilon)$ iterations. Each iteration consists of up to $K-1$ messages through the matrices $\mathbf{M}_k(t)$ whose dimensions are equal to the cardinalities of S_k for $k = 2, \ldots, K$. A simple choice for the bounds is $L = 0$ since $\mathbf{K}$ is positive definite, and

$$U = \min_{k=1,\ldots,K}\{\mathrm{eig}_{\min}(\mathbf{K}_{C_k,C_k})\} \tag{56}$$

as proved in Lemma 2 in the Appendix.

Algorithm 2: Bisection line search for DPCA

```
Input: $\mathbf{K}$, $L$, $U$, $\epsilon$, clique tree structure
Output: $t$
while $U - L > \epsilon$ do
    $t = (U + L)/2$
    $\mathbf{Q} = \mathbf{K}$
    for $k = K, \cdots, 2$ do
        if $t < \mathrm{eig}_{\min}(\mathbf{Q}_{R_k,R_k})$ then
            $\mathbf{M}_k(t) = \mathbf{Q}_{S_k,R_k}(\mathbf{Q}_{R_k,R_k} - t\mathbf{I})^{-1}\mathbf{Q}_{R_k,S_k}$
            $\mathbf{Q}_{S_k,S_k} = \mathbf{Q}_{S_k,S_k} - \mathbf{M}_k(t)$
        else
            $U = t$
            break loop
        end
    end
    if $U > t$ then
        if $t < \mathrm{eig}_{\min}(\mathbf{Q}_{C_1,C_1})$ then
            $L = t$
        else
            $U = t$
        end
    end
end
```

Given a principal eigenvalue λ, its corresponding eigenvector can be computed by solving $\mathbf{Qu} = \mathbf{0}$ where $\mathbf{Q} = \mathbf{K} - \lambda\mathbf{I}$. Similarly to Section II-D, we begin with $k = K$ and partition H_k into R_k and H_{k-1}. We test the singularity of $\mathbf{Q}_{R_k,R_k}$. If it is singular, then λ is associated with R_k. Otherwise, we send the message $\mathbf{M}_k(\lambda)$ to $\mathbf{H}_{k-1}$ and repartition it. We continue until we find the associated remainder R_k or reach the first clique. Then, we compute the corresponding local null vector and begin propagating it to the higher remainders as expressed in (27). A pseudo code of this method is provided in Algorithm 2.

Algorithm 2 can be easily extended to compute higher order eigenvalues through application of Proposition 2. For this purpose, note that the matrix in (39) has the same structure as (36) and therefore can be recursively partitioned again. The only difference is that the rank of the modification is increased at each clique and requires larger message matrices. Thus, the algorithm is efficient as long as the size of the separators $(|S_k|)$, the number of cliques (K) and the number of required eigenvalues (j) are all relatively small in comparison to p. Given any eigenvalue (first or high order), Algorithm 3 requires one backward and one forward sweep through the cliques in order the compute its associated eigenvector.

Algorithm 3: Eigenvector computation

```
Input: $\mathbf{Q}$, clique tree structure
Output: $\mathbf{u}$
$\mathbf{u} = \mathbf{0}$
$\mathbf{Q} = \mathbf{K}$
$k = K$
while $(k > 1)$ & $(\mathbf{Q}_{R_k,R_k}$ non singular$)$ do
    $\mathbf{M}_k = \mathbf{Q}_{S_k,R_k}\mathbf{Q}^{-1}_{R_k,R_k}\mathbf{Q}_{R_k,S_k}$
    $\mathbf{Q}_{S_k,S_k} = \mathbf{Q}_{S_k,S_k} - \mathbf{M}_k$
    $k = k - 1$
end
$\mathbf{u}(C_k) = \mathbf{u}_{\mathrm{null}}(\mathbf{Q}_{C_k,C_k})$
for $k = k + 1, \cdots, K$ do
    $\mathbf{u}(R_k) = -\mathbf{Q}^{-1}_{R_k,R_k}\mathbf{Q}_{R_k,S_k}\mathbf{u}(S_k)$
end
```

IV. Synthetic Tracking Example

We now illustrate the performance of DPCA using a synthetic numerical example. Specifically, we use DPCA to track the first principle component in a slowly time varying setting. We define a simple graphical model with 305 nodes representing three fully connected networks with only 5 coupling nodes, i.e., $C_1 = \{1, \ldots, 100, 301, \ldots, 305\}$, $C_2 = \{101, \ldots, 200, 301, \ldots, 305\}$, and $C_3 = \{201, \ldots, 300, 301, \ldots, 305\}$. We generate 5500 length $p = 305$ vectors $\mathbf{x}_i$ of zero mean, unit variance and independent Gaussian random variables. At each time point, we define $\mathbf{K}$ through (46) using a sliding window of $n = 500$ realizations with 400 samples overlap. Next, we run DPCA using Algorithm 1. Due to slow time variation, we define the lower (L) and upper (U) bounds as the value of the previous time point minus and plus 0.1, respectively. We define the tolerance as $\epsilon = 0.001$ corresponding to 8 iterations. Fig. 2 shows the exact value of the minimal eigenvalue as a function of time along with its DPCA estimates at the $4'$th, $6'$th and $8'$th iterations. It is easy to see that a few iterations suffice for tracking the maximal eigenvalue at high accuracy. Each iteration involves three EVDs of approximately 105×105 matrices and communication through two messages of size 5×5. For comparison, a centralized solution would require

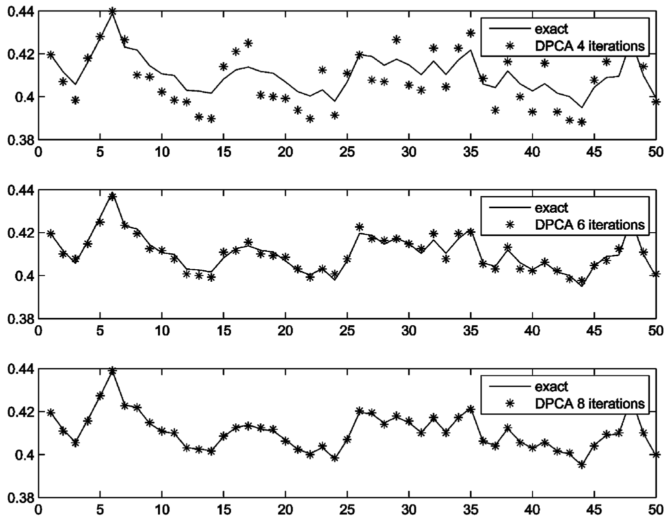

Fig. 2. Iterations of the DPCA bisection line-search in a time varying scenario.

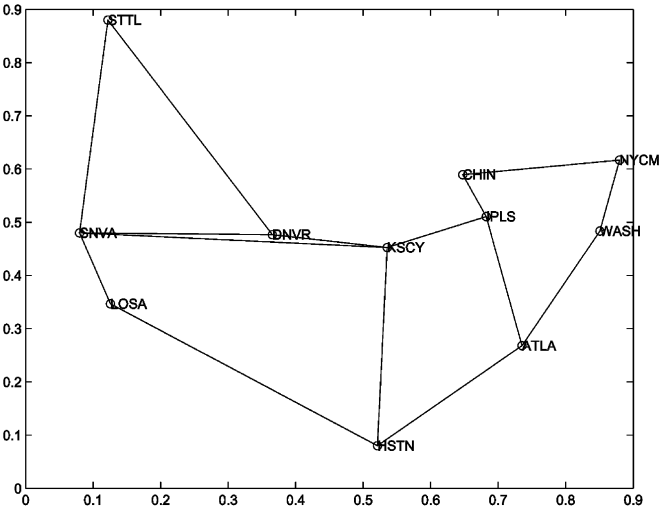

Fig. 3. Map of the Abilene network.

sending a set of 100 length 305 vectors to a central processing unit which computes an EVD of a matrix of size 305 × 305.

V. Application to Distributed Anomaly Detection

A promising application for DPCA is distributed anomaly detection in computer networks. In this context, PCA is used for learning a low dimensional model for normal behavior of the traffic in the network. The samples are projected into the subspace associated with the first principal components. Anomalies are then easily detected by examining the residual norm. Our hypothesis is that the connectivity map of the network is related to its statistical graphical model. The intuition is that two distant links in the network are (approximately) independent conditioned on the links connecting them and therefore define a graphical model. We do not rigorously support this claim but rather apply it in a heuristic manner in order to illustrate DPCA.

Following [12], [15], we consider a real world dataset of Abilene, the Internet2 backbone network. This network carries traffic from universities in the United States. Fig. 3 shows its connectivity map consisting of 11 routers and 41 links (each line corresponds to two links and there are additional links from each of the nodes to itself). To avoid confusion, we emphasize that the 41 nodes of the conditional independence graphs proposed below are the Abilene links and not the routers as may be implied from Fig. 3.

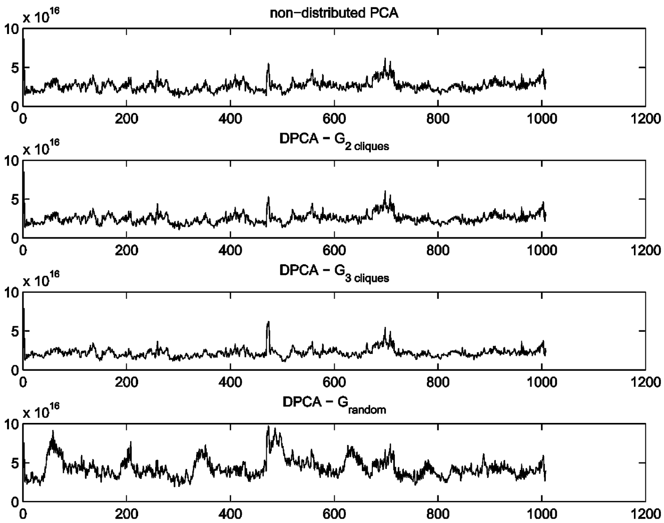

Fig. 4. Projection into anomaly subspace with and without graphical models.

Our first proposed graph, denoted by $\mathcal{G}_{\text{2cliques}}$, consists of two cliques: an eastern clique consisting of the separating links DNVR-KSCY, SNVA-KSCY, and LOSA-HSTN and the links to their east, and a western clique consisting of the separating links and the links to their west. Unlike the topology of Fig. 3, all the nodes are connected within each clique of our model. This graph corresponds to a decomposable concentration matrix with a sparsity level of 0.33. Our second proposed graph, denoted by $\mathcal{G}_{\text{3cliques}}$, is obtained by redividing the eastern clique again into two cliques separated through four coupling links: IPLS-CHIN and ATLA-WASH. Its sparsity level is 0.43. Finally, for comparison we randomly generate an arbitrary graph $\mathcal{G}_{\text{random}}$ over the Abilene nodes, with an identical structure as $\mathcal{G}_{\text{3cliques}}$ (three cliques of the same cardinalities), which is not associated with the topology of the Abilene network.

In our experiments, we learn the 41 × 41 covariance matrix from a 41 × 1008 data matrix representing 1008 samples of the traffic on each of the 41 Abilene links during April 7–13, 2003. We compute PCA and project each of the 1008 samples of dimension 41 into the null space of the first four principal components. The norm of these residual samples is plotted in the top plot of Fig. 4. It is easy to see the spikes putatively associated with anomalies. Next, we examine the residuals using DPCA with $\mathcal{G}_{\text{2cliques}}$, $\mathcal{G}_{\text{3cliques}}$ and $\mathcal{G}_{\text{random}}$. The norms of the residuals are plotted in the three lower plots of Fig. 4, respectively. As expected, the topology based plots are quite similar with spikes occurring at the times of these anomalies. Thus, we conclude that the decomposable graphical models for Abilene are a good approximation and do not cause substantial loss of information (at least for the purpose of anomaly detection). On the other hand, the residual norm using the random graph is a poor approximation as it does not preserve the anomalies detected by the full nondistributed PCA. These conclusions are supported

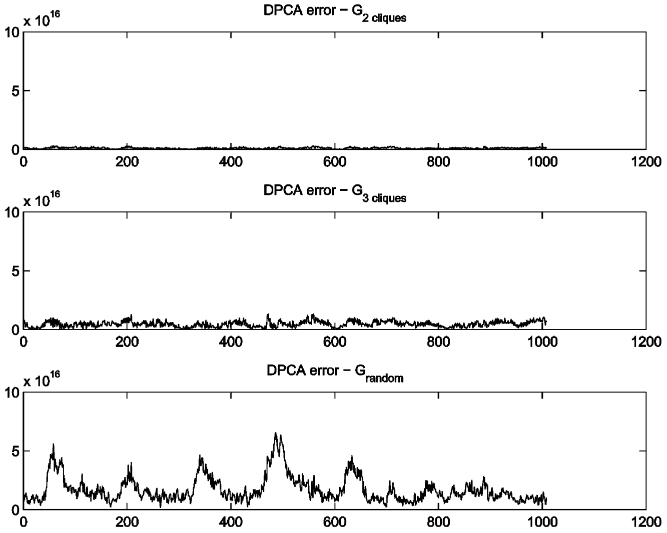

Fig. 5. Absolute error in projection into anomaly subspace with different graphical models.

in Fig. 5 where we show the absolute errors of DPCA with respect to PCA using the different graphical models. It is easy to see that $\mathcal{G}_{\rm 2cliques}$ results in minimal error, $\mathcal{G}_{\rm 3cliques}$ provides a reasonable tradeoff between performance and computational complexity (through its increased sparsity level), while graph $\mathcal{G}_{\rm random}$ is clearly a mismatched graphical model and results in significant increase in error.

VI. Discussion and Future Work

In this paper, we introduced DPCA and derived a decentralized method for its computation. We proposed distributed anomaly detection in communication networks as a motivating application for DPCA and investigated possible graphical models for such settings.

Future work should examine the statistical properties of DPCA. From a statistical perspective, DPCA is an extension of classical PCA to incorporate additional prior information. Thus, it would be interesting to analyze the distribution of its components and quantify their significance, both under the true graphical model and under mismatched models. In addition, DPCA is based on the intimate relation between the inverse covariance and the conditional Gaussian distribution. Therefore, it will also be important to assess its sensitivity to non-Gaussian sources. Finally, alternative methods to ML in singular and ill conditioned scenarios should be considered.

Another interesting extension of DPCA is its generalization to nondecomposable graphical models. Maximum likelihood estimation in nondecomposable models does not have a closed form solution but can still be implemented using the iterative proportional fitting algorithm [13], [16]. Future work could focus on similar iterative methods for eigenvalue computations in arbitrary graphs.

Appendix

Lemma 2: Let $\mathbf{K}$ be a symmetric matrix, and let a be a subset of its indices. Then, $\mathrm{eig}_{\min}(\mathbf{K}) \leq \mathrm{eig}_{\min}\left([\mathbf{K}]_{a,a}\right)$.

Proof: For simplicity, we assume that a is the subset of the first $|a|$ indices. The proof is a simple application of the Rayleigh quotient characterization of the minimal eigenvalues:

$$\begin{aligned} \mathrm{eig}_{\min}(\mathbf{K}) &= \min_{\mathbf{u}} \frac{\mathbf{u}^T\mathbf{K}\mathbf{u}}{\mathbf{u}^T\mathbf{u}} && (57)\\ &\leq \frac{[\mathbf{v}^T \quad \mathbf{0}^T]\,\mathbf{K}\begin{bmatrix}\mathbf{v}\\ \mathbf{0}\end{bmatrix}}{[\mathbf{v}^T \quad \mathbf{0}^T]\begin{bmatrix}\mathbf{v}\\ \mathbf{0}\end{bmatrix}} && (58)\\ &= \frac{\mathbf{v}^T\mathbf{K}_{a,a}\mathbf{v}}{\mathbf{v}^T\mathbf{v}} && (59)\\ &= \min_{\mathbf{u}} \frac{\mathbf{u}^T\mathbf{K}_{a,a}\mathbf{u}}{\mathbf{u}^T\mathbf{u}} && (60)\\ &= \mathrm{eig}_{\min}\{\mathbf{K}_{a,a}\} && (61)\end{aligned}$$

where $\mathbf{v}$ is the optimal solution to (60). ■

Lemma 3: Let $\mathbf{X} = \begin{bmatrix}\mathbf{X}_{11} & \mathbf{X}_{12}\\ \mathbf{X}_{21} & \mathbf{X}_{22}\end{bmatrix}$ be a positive semidefinite matrix, and let $\mathbf{u}$ be a vector in the null space of $\mathbf{X}_{22}$. Then, $\mathbf{u}$ is also in the null space of $\mathbf{X}_{12}$.

Proof: Due to the semidefiniteness, we can decompose $\mathbf{X}$ as

$$\begin{bmatrix}\mathbf{X}_{11} & \mathbf{X}_{12}\\ \mathbf{X}_{21} & \mathbf{X}_{22}\end{bmatrix} = \begin{bmatrix}\mathbf{V}_1\\ \mathbf{V}_2\end{bmatrix}[\mathbf{V}_1^T \quad \mathbf{V}_2^T]. \qquad (62)$$

Therefore, $\mathbf{X}_{22} = \mathbf{V}_2\mathbf{V}_2^T$ and $\mathbf{V}_2^T\mathbf{u} = \mathbf{0}$. On the other hand, $\mathbf{X}_{12} = \mathbf{V}_1\mathbf{V}_2^T$ and

$$\mathbf{X}_{12}\mathbf{u} = \mathbf{V}_1\mathbf{V}_2^T\mathbf{u} = \mathbf{V}_1\mathbf{0} = \mathbf{0} \qquad (63)$$

as required.

Acknowledgment

The authors would like to thank C. Scott for providing the Abilene data, A. T. Puig for stimulating discussions, and the anonymous referees for their constructive suggestions.

References

[1] T. W. Anderson*, An Introduction to Multivariate Statistical Analysis*, Second ed. New York: Wiley, 1971.

[2] Z. J. Bai, R. H. Chan, and F. T. Luk*, Advanced Parallel Processing Technologies*. Berlin, Germany: Springer, 2005, ch. Principal Component Analysis for Distributed Data Sets with Updating, pp. 471–483.

[3] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation," *J. Mach. Learn. Res.*, vol. 9, pp. 485–516, Mar. 2008.

[4] S. Boyd and L. Vandenberghe*, Introduction to Convex Optimization With Engineering Applications*. La Jolla, CA: Stanford University Press, 2003.

[5] M. Cetin, L. Chen, J. W. Fisher, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks: A graphical models perspective," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 42–55, Jul. 2006.

[6] P. Chhabra, C. Scott, E. Kolaczyk, and M. Crovella, "Distributed spatial anomaly detection," in *Proc. INFOCOM*, Apr. 2008.

[7] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, pp. 157–175, 1972.

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the LASSO," *Biostat*, vol. 9, no. 3, pp. 432–441, Jul. 2008.

[9] M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed Karhunen Loeve transform," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5177–5196, Dec. 2006.

[10] G. H. Golub and C. F. Van Loan*, Matrix Computations*. Baltimore, MD: John Hopkins University Press, 1983.

[11] T. Hastie, R. Tibshirani, and J. Friedman*, The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2001.

[12] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. D. Joseph, and N. Taft, "In-network PCA and anomaly detection," in *Proc. NIPS'2006*, Dec. 2006.

[13] M. I. Jordan*, Introduction to Graphical Models*, 2008, submitted for publication.

[14] H. Kargupta, W. Huang, K. Sivakumar, and E. Hohnson, "Distributed clustering using collective principal component analysis," *Knowl. Inf. Syst.*, vol. 3, no. 4, pp. 422–448, Nov. 2001.

[15] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 219–230, 2004.

[16] S. L. Lauritzen*, Graphical Models*. New York: Oxford, 1996, vol. 17, pt. Oxford Statistical Science Series.

[17] Y. Qu, G. Ostrouchovz, N. Samatovaz, and A. Geist, "Principal component analysis for dimensions reduction in massive distributed data sets," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2002.

[18] O. Roy and M. Vetterli, "Dimensionality reduction for distributed estimation in the infinite dimensional regime," *IEEE Trans. Inf. Theory*, vol. 54, no. 2, pp. 1655–1669, Apr. 2008.

[19] I. D. Schizas, G. B. Giannakis, and Z. Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007.

[20] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comp.*, vol. 13, no. 10, pp. 2173–2200, 2001.

[21] J. J. Xiao, A. Ribeiro, Z. Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, Jul. 2006.

[22] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[23] Y. Zhu, E. Song, J. Zhou, and Z. You, "Optimal dimensionality reduction of sensor data in multisensor estimation fusion," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1631–1639, May 2005.

Ami Wiesel (S'02–M'09) received the B.Sc. and M.Sc. degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, Israel, in 2000 and 2002, respectively, and the Ph.D. degree in electrical engineering from the Technion–Israel Institute of Technology, Haifa, in 2007.

Currently, he is a Postdoctoral Fellow at the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor.

Dr. Wiesel received the Young Author Best Paper Award for a 2006 paper in the IEEE Transactions in Signal Processing and a Student Paper Award for the 2005 Workshop on Signal Processing Advances in Wireless Communications (SPAWC) paper. He was awarded the Weinstein Study Prize in 2002, the Intel Award in 2005, the Viterbi Fellowship in 2005 and 2007, and the Marie Curie Fellowship in 2008.

Alfred O. Hero (S'79–M'84–SM'96–F'98) received the B.S. (*summa cum laude*) from Boston University, MA, in 1980 and the Ph.D. degree from Princeton University, Princeton, NJ, in 1984, both in electrical engineering.

Since 1984, he has been with the University of Michigan, Ann Arbor, where he is the R. Jamison and Betty Professor of Engineering. His primary appointment is in the Department of Electrical Engineering and Computer Science and he also has appointments, by courtesy, in the Department of Biomedical Engineering and the Department of Statistics. In 2008 he was awarded the the Digiteo Chaire d'Excellence, sponsored by Digiteo Research Park in Paris, located at the Ecole Superieure d'Electricite, Gif-sur-Yvette, France. He has held other visiting positions at LIDS Massachussets Institute of Technology (2006), Boston University (2006), I3S University of Nice, Sophia-Antipolis, France (2001), Ecole Normale Supérieure de Lyon (1999), Ecole Nationale Supéerieure des Téeléecommunications, Paris (1999), Lucent Bell Laboratories (1999), Scientific Research Labs of the Ford Motor Company, Dearborn, Michigan (1993), Ecole Nationale Superieure des Techniques Avancees (ENSTA), Ecole Superieure d'Electricite, Paris (1990), and M.I.T. Lincoln Laboratory (1987–1989). His recent research interests have been in detection, classification, pattern analysis, and adaptive sampling for spatio–temporal data. Of particular interest are applications to network security, multimodal sensing and tracking, biomedical imaging, and genomic signal processing.

Dr. Hero is a member of Tau Beta Pi, the American Statistical Association (ASA), the Society for Industrial and Applied Mathematics (SIAM), and the U.S. National Commission (Commission C) of the International Union of Radio Science (URSI). He has received a IEEE Signal Processing Society Meritorious Service Award (1998), IEEE Signal Processing Society Best Paper Award (1998), a IEEE Third Millenium Medal and a 2002 IEEE Signal Processing Society Distinguished Lecturership. He was President of the IEEE Signal Processing Society (2006–2007) and during his term served on the TAB Periodicals Committee (2006). He was a member of the IEEE TAB Society Review Committee (2008) and is Director-elect of IEEE for Division IX (2009).