Decomposable Principal Component Analysis

Ami Wiesel and Alfred O. Hero III

Department of Electrical Engineering and Computer Science

University of Michigan, Ann Arbor, MI 48109, USA

E-mails: {amiw,hero}@umich.edu

Abstract

We consider principal component analysis (PCA) in decomposable Gaussian graphical models. We exploit the prior information in these models in order to distribute its computation. For this purpose, we reformulate the problem in the sparse inverse covariance (concentration) domain and solve the global eigenvalue problem using a sequence of local eigenvalue problems in each of the cliques of the decomposable graph. We demonstrate the application of 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.

I. INTRODUCTION

We consider principal component analysis (PCA) in Gaussian graphical models. PCA is a classical dimensionality reduction method which is frequently used in statistics and machine learning [11], [1]. The first principal components of a multivariate are its orthogonal linear combinations which preserve most of the variance. In the Gaussian case, PCA has special properties which make it especially favorable: it is the best linear approximation of the data and it provides independent components. On the other hand, Gaussian graphical models, also known as covariance selection models, provide a graphical representation of the conditional independence structure within the Gaussian distribution [16], [7]. Exploiting the extensive knowledge and literature on graph theory, graphical models allow for efficient distributed implementation of statistical inference algorithms, e.g., the well known belief propagation method and the junction tree algorithm [20], [13]. In particular, decomposable graphs, also known as chordal or triangulated graphs, provide simple and intuitive inference methods due to their appealing structure. 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 [23], [19], [21], [9], [18], and decentralized data mining techniques [14], [2], [17]. It is also used in anomaly detection in computer networks [15], [6], [12]. In particular, [9], [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 tolerance. On the other hand, it assumes additional prior knowledge in the form of a graphical model which previous works did not take into account. Although, it is interesting to note that the Gauss Markov source example in [9], [18] is probably the most celebrated decomposable graphical model. Therefore, we now address the availability of such prior information. In general, practical applications do not necessarily satisfy any obvious conditional independence structure. In such scenarios, DPCA can be interpreted as an approximate PCA method that allows a tradeoff between accuracy and decentralization by introducing sparsity. In other problems it is reasonable to assume that an unknown structure exists and can be learned from the observed data using existing methods such as [3], [8], [22]. Alternatively, a graphical model can be derived from non-statistical prior knowledge on the specific application. An intuitive example is distributed networks in which the topology of the network suggests a statistical graph as exploited in [5]. Finally, we emphasize that even if a prior graphical model is available, it does not necessarily satisfy a decomposable form. In this case, a decomposable approximation can be obtained using classical graph theory algorithms [13].

PCA can be interpreted as maximum likelihood (ML) estimation of the covariance using the available data followed by its eigenvalue decomposition. When a prior graphical model is available, PCA can still be easily obtained by adjusting the ML estimation phase to incorporate the prior conditional independence structure using existing methods [16], [7], and then computing the eigenvalue decomposition (EVD). The drawback to this approach is that it does not exploit the structure of the graph in the EVD phase. This disadvantage is the primary motivation to DPCA which is specifically designed to utilize the structure of Gaussian graphical models. Decomposable covariance selection models result in sparse concentration (inverse covariance) matrices which can be estimated in a decentralized manner. Therefore, we propose to reformulate DPCA in the concentration domain and solve the global EVD using a sequence of local EVD problems in each of the cliques of the decomposable graph with a small amount of message passing.

This allows for distributed implementation according to the topology of the graph and reduces the need to collect all the observed data in a centralized processing unit. When the 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 [15], [12]. In this context, DPCA learns a low dimensional model of the normal traffic behavior and allows for simple outlier detection. This application is natural since the network's topology provides a physical basis for constructing an approximate a graphical model. For example, consider two nodes which are geographically distant and linked only through a long path of nodes. It is reasonable to believe that these two sensors are independent conditioned on the path, but a theoretical justification of this assertion is difficult and depends on the specific problem formulation. We examine the validity of this claim in the context of anomaly detection in the Abilene network using a real-world dataset. 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 easy to explain using a special graph of two cliques which is their main building block. Therefore, we begin in section II by introducing the problem formulation and solution to DPCA in this simple case. The generalization to decomposable graphs is presented in section III which consists of their technical definitions followed by a recursive application of the two cliques solution. We demonstrate the use of DPCA using two numerical 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 transpose and matrix inverse, respectively. The cardinality of a set a is denoted by $|a|$. 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 sub-vectors or sub-matrices, respectively, and $[\mathbf{X}]_{a,:}$ denotes the sub-matrix 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 be the building block for the general algorithm.
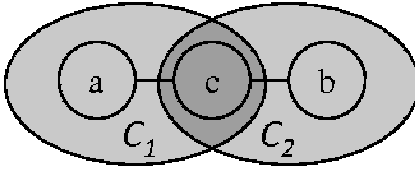
Fig. 1. Graphical model with two cliques modeling a 3 node network in which a and b are conditionally independent given c.

A. Problem Formulation

Let $\mathbf{x} = \begin{bmatrix} \mathbf{x}_a^T \ \mathbf{x}_c^T \ \mathbf{x}_b^T \end{bmatrix}^T$ be a length p, zero mean Gaussian random vector in which $[\mathbf{x}]_a$ and $[\mathbf{x}]_b$ are independent conditionally on $[\mathbf{x}]_c$ where a, c and b are disjoint subsets of indices. For later use, we use graph terminology and define two cliques of indices $C_1 = \{a, c\}$ and $C_2 = \{c, b\}$ coupled through the separator $S = \{c\}$ (see Fig. 1). We assume that the covariance matrix of $\mathbf{x}$ is unknown, but the conditional independence structure (defined through index sets C_1 and C_2) is known.

The input to DPCA is a set of n independent and identically distributed realizations of $\mathbf{x}$ denoted by $\mathbf{x}_i$ for $i = 1, \cdots, n$. More specifically, this input is distributed in the sense that the first clique has access to $[\mathbf{x}_i]_{C_1}$ for $i = 1, \cdots, n$, whereas the second clique has access only to $[\mathbf{x}_i]_{C_2}$ for $i = 1, \cdots, n$. Using this data and minimal 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 sub-vectors $[\mathbf{u}]_{C_1}$ 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. Due to the known conditional independence structure, the ML estimate has a simple closed form solution which can be computed in a distributed manner (more details about this procedure can be found in [16]). Each clique and the separator computes their own local sample covariance matrices

$$\tilde{\mathbf{S}}^{C_1, C_1} = \frac{1}{n} \sum_{i=1}^{n} [\mathbf{x}_i]_{C_1} [\mathbf{x}_i]_{C_1}^T \tag{1}$$

$$\tilde{\mathbf{S}}^{C_2, C_2} = \frac{1}{n} \sum_{i=1}^{n} [\mathbf{x}_i]_{C_2} [\mathbf{x}_i]_{C_2}^T \tag{2}$$

$$\tilde{\mathbf{S}}^{S,S} = \frac{1}{n}\sum_{i=1}^{n}[\mathbf{x}_i]_S[\mathbf{x}_i]_S^T, \tag{3}$$

where the tilde and the superscripts are used to emphasize that these are local estimates. Similarly, the local concentration matrices, also known as precision matrices, are defined as

$$\tilde{\mathbf{K}}^{C_1,C_1} = \left(\tilde{\mathbf{S}}^{C_1,C_1}\right)^{-1} \tag{4}$$

$$\tilde{\mathbf{K}}^{C_2,C_2} = \left(\tilde{\mathbf{S}}^{C_2,C_2}\right)^{-1} \tag{5}$$

$$\tilde{\mathbf{K}}^{S,S} = \left(\tilde{\mathbf{S}}^{S,S}\right)^{-1}, \tag{6}$$

where it is assumed that the matrices are non-singular (otherwise, the ML estimate does not exist). Next, the global ML concentration matrix $\mathbf{K}$ is obtained by requiring

$$\mathbf{K}_{a,b} = \mathbf{K}_{b,a}^T = \mathbf{0} \tag{7}$$

due to the conditional independence of $\mathbf{x}_a$ and $\mathbf{x}_b$ given $\mathbf{x}_c$. The global solution is

$$\mathbf{K} = \begin{bmatrix} \tilde{\mathbf{K}}^{C_1,C_1} & & \mathbf{0} \\ & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & & \\ \mathbf{0} & & \tilde{\mathbf{K}}^{C_2,C_2} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}^{S,S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{8}$$

It is easy to see that the sub-matrices associated with the cliques are perturbations of to their local versions:

$$[\mathbf{K}]_{C_1,C_1} = \tilde{\mathbf{K}}^{C_1,C_1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_b \end{bmatrix} \tag{9}$$

$$[\mathbf{K}]_{C_2,C_2} = \tilde{\mathbf{K}}^{C_2,C_2} + \begin{bmatrix} \mathbf{M}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{10}$$

and require only message passing via $\mathbf{M}_b$ and $\mathbf{M}_a$:

$$\mathbf{M}_b = \left[\tilde{\mathbf{K}}^{C_2,C_2}\right]_{S,S} - \tilde{\mathbf{K}}^{S,S} \tag{11}$$

$$\mathbf{M}_a = \left[\tilde{\mathbf{K}}^{C_1,C_1}\right]_{S,S} - \tilde{\mathbf{K}}^{S,S}. \tag{12}$$

The dimension of these messages is equal to $|S|$ which is presumably small. Thus, the global ML concentration matrix can be easily found in a distributed manner.

The global covariance estimate is simply defined as the inverse of its concentration $\mathbf{S} = \mathbf{K}^{-1}$. It is consistent with the local estimates of its sub-matrices:

$$[\mathbf{S}]_{C_1,C_1} = \tilde{\mathbf{S}}^{C_1,C_2} \tag{13}$$

$$[\mathbf{S}]_{C_2,C_2} = \tilde{\mathbf{S}}^{C_2,C_2}, \tag{14}$$

but there is no special intuition regarding its $[\mathbf{S}]_{a,b}$ and $[\mathbf{S}]_{b,a}$ sub-blocks.

C. Solution: first principal eigenvalue

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

$$\mathbf{u}^T\mathbf{S}\mathbf{u}, \tag{15}$$

which is maximized subject to a norm constraint to yield

$$\text{eig}_{\max}(\mathbf{S}) = \begin{cases} \max_{\mathbf{u}} & \mathbf{u}^T\mathbf{S}\mathbf{u} \\ \text{s.t.} & \mathbf{u}^T\mathbf{u} = 1. \end{cases} \tag{16}$$

This optimization gives both the maximal eigenvalue of $\mathbf{S}$ and the 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 $\mathbf{S}$ 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 minimal message passing in order to calculate its local version of $\text{eig}_{\max}(\mathbf{S})$ 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

$$\text{eig}_{\max}(\mathbf{S}) = \frac{1}{\text{eig}_{\min}(\mathbf{K})}, \tag{17}$$

when the inverse $\mathbf{K} = \mathbf{S}^{-1}$ exists. The corresponding eigenvectors are also identical. The advantage of working with $\mathbf{K}$ instead of $\mathbf{S}$ is that we can directly exploit $\mathbf{K}$'s sparsity as expressed in (7).

Before continuing it is important to address the question of singularity of $\mathbf{S}$. One may claim that working in the concentration domain is problematic since $\mathbf{S}$ 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]. In fact, the ML covariance is defined as the inverse of its concentration, and thus the issue of singularity is an intrinsic problem of ML estimation rather than the DPCA solution.

We now return to the problem of finding

$$\lambda = \text{eig}_{\min}(\mathbf{K}) \tag{18}$$

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

$$\lambda = \sup \quad t \quad \text{s.t.} \quad t < \text{eig}_{\min}(\mathbf{K}) \tag{19}$$

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 $\mathrm{eig}_{\min}\left(\mathbf{K}\right)$ 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 $\mathbf{K}_{a,b} = \mathbf{K}_{b,a}^T = \mathbf{0}$. Then, the constraint

$$t < \mathrm{eig}_{\min}\left(\begin{bmatrix} \mathbf{K}_{a,a} & \mathbf{K}_{a,c} & \mathbf{0} \\ \mathbf{K}_{c,a} & \mathbf{K}_{c,c} & \mathbf{K}_{c,b} \\ \mathbf{0} & \mathbf{K}_{b,c} & \mathbf{K}_{b,b} \end{bmatrix}\right) \tag{20}$$

is equivalent to the following pair of constraints

$$t \quad < \quad \mathrm{eig}_{\min}\left(\mathbf{K}_{b,b}\right) \tag{21}$$

$$t \quad < \quad \mathrm{eig}_{\min}\left(\mathbf{K}_{C_1,C_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(t\right) \end{bmatrix}\right) \tag{22}$$

with the *message matrix* defined as

$$\mathbf{M}\left(t\right) = \mathbf{K}_{c,b}\left(\mathbf{K}_{b,b} - t\mathbf{I}\right)^{-1}\mathbf{K}_{b,c}. \tag{23}$$

Proof: The proof is obtained by rewriting (20) as a linear matrix inequality

$$\begin{bmatrix} \mathbf{K}_{a,a} & \mathbf{K}_{a,c} & \mathbf{0} \\ \mathbf{K}_{c,a} & \mathbf{K}_{c,c} & \mathbf{K}_{c,b} \\ \mathbf{0} & \mathbf{K}_{b,c} & \mathbf{K}_{b,b} \end{bmatrix} - t\mathbf{I} \succ \mathbf{0} \tag{24}$$

and decoupling this inequality using the following lemma:

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

$$\mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}. \tag{25}$$

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

Applying Schur's Lemma to (24) with $\mathbf{A} = \mathbf{K}_{C_1,C_1}$ and rearranging yields

$$t\mathbf{I} \quad \prec \quad \mathbf{K}_{b,b} \tag{26}$$

$$t\mathbf{I} \quad \prec \quad \mathbf{K}_{C_1,C_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(t\right) \end{bmatrix}. \tag{27}$$

Finally, (21) and (22) are obtained by rewriting (26) and (27) as eigenvalue inequalities, respectively. ∎

Proposition 1 provides an intuitive distributed solution to (19). 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 .

We note that the solution in Proposition 1 is asymmetric with respect to the cliques. The global constraint is replaced by two local constraints regarding clique $C_1 = \{a, c\}$ and the remainder $\{b\}$. Alternatively, we can exchange the order and partition the indices into $\{a\}$ and $C_2 = \{c, b\}$. This asymmetry will become important in the next section when we extend the results to general decomposable graphs.

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 $\mathbf{u} = \mathbf{u}_{\text{null}}(\mathbf{Q})$. The matrix $\mathbf{Q}$ follows the same block sparse structure as $\mathbf{K}$, and the linear set of equations $\mathbf{Q}\mathbf{u} = \mathbf{0}$ can be solved in a distributed manner. There are two possible solutions. Usually, $\mathbf{Q}_{bb}$ is non-singular in which case the solution is

$$[\mathbf{u}]_{C_1} = \mathbf{u}_{\text{null}}\left(\mathbf{Q}_{C_1,C_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}\right) \tag{28}$$

$$[\mathbf{u}]_b = -\mathbf{Q}_{b,b}^{-1}\mathbf{Q}_{b,c}[\mathbf{u}]_c, \tag{29}$$

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

$$\mathbf{M} = \mathbf{Q}_{c,b}\mathbf{Q}_{b,b}^{-1}\mathbf{Q}_{b,c}. \tag{30}$$

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

$$[\mathbf{u}]_{C_1} = \mathbf{0} \tag{31}$$

$$[\mathbf{u}]_b = \mathbf{u}_{\text{null}}(\mathbf{K}_{b,b}). \tag{32}$$

This singular case is highly unlikely as the probability of (31) in continuous models is zero. However, it should be checked for completeness.

E. Solution: higher order components

In practice, dimensionality reduction involves the projection of the data into the subspace of a few of the first principal components. We now show that the algorithm in II-C can be extended to provide higher order components.

The j'th principal component is defined as the linear transformation which is orthogonal to the preceding components and preserves maximal variance. Similarly to the first component it is given by $\mathbf{X}_j = \mathbf{u}_j^T \mathbf{x}$ where $\mathbf{u}_j$ is the j'th principal eigenvector of $\mathbf{S}$. In the concentration domain, $\mathbf{u}_j$ is the eigenvector associated with λ_j, the j'th smallest eigenvalue of $\mathbf{K}$.

In order to distribute the computation of λ_j, we adjust (19) using the following lemma:

Lemma 2: Let $\mathbf{K}$ be a symmetric matrix with eigenvalues $\lambda_1 \leq, \cdots, \leq \lambda_p$ and eigenvectors $\mathbf{u}_1, \cdots, \mathbf{u}_p$. Then,

$$\lambda_j = \sup_{\{v_i\}_{i=1}^{j-1}, t} t \quad \text{s.t.} \quad t < \text{eig}_{\min}\left(\mathbf{K} + \sum_{i=1}^{j-1} v_i \mathbf{u}_i \mathbf{u}_i^T\right). \tag{33}$$

The optimal v_i are any values which satisfy $v_i > \lambda_j - \lambda_i$ for $i = 1, \cdots, j-1$.

Proof: The proof is based on the recursive variational characterization of of the j'th smallest eigenvalue[1]:

$$\lambda_j = \begin{cases} \min_{\mathbf{u}} & \mathbf{u}^T \mathbf{K} \mathbf{u} \\ \text{s.t.} & \mathbf{u}^T \mathbf{u} = 1 \\ & \mathbf{u}^T \mathbf{u}_i = 0, \quad i = 1, \cdots, j-1 \end{cases} \tag{34}$$

where $\mathbf{u}_i$ are the preceding eigenvectors, and the optimal solution $\mathbf{u}_j = \mathbf{u}$ is the eigenvector associated with λ_j. A dual representation can be obtained using Lagrange duality. We rewrite the orthogonality restrictions as quadratic constraints $\mathbf{u}^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u} = 0$ and eliminate them using Lagrange multipliers:

$$\lambda_j \geq \max_{t, \{v_i\}_{i=1}^{j-1}} \min_{\mathbf{u}} t + \mathbf{u}^T \left[\mathbf{K} - t\mathbf{I} + \sum_{i=1}^{j-1} v_i \mathbf{u}_i \mathbf{u}_i^T\right] \mathbf{u} \tag{35}$$

where the inequality is due to the weak duality [4]. The inner minimization is unbounded unless

$$\mathbf{K} - t\mathbf{I} + \sum_{i=1}^{j-1} v_i \mathbf{u}_i \mathbf{u}_i^T \succeq \mathbf{0}. \tag{36}$$

Therefore,

$$\lambda_j \geq \max_{t, \{v_i\}_{i=1}^{j-1}} \quad \text{s.t.} \quad t \leq \text{eig}_{\min}\left(\mathbf{K} + \sum_{i=1}^{j-1} v_i \mathbf{u}_i \mathbf{u}_i^T\right) \tag{37}$$

Lagrange duality does not guaranty an equality in (37) since (34) is not convex. However, it is easy to see that the inequality is tight and can be attained by choosing $\mathbf{u} = \mathbf{u}_j$. Finally, (33) is obtained by replacing the maximum with a supremum and relaxing the constraint. ∎

---

[1]There is also a non-recursive characterization known as Courant-Fischer theorem which results in a similar maximin representation [10].

Lemma 2 allows us to find λ_j in a distributed manner. We replace $\mathbf{K}$ with $\overline{\mathbf{K}} = \mathbf{K} + \mathbf{U}\mathbf{D}\mathbf{U}^T$ where $\mathbf{U}$ is a $p \times (j-1)$ matrix with the preceding eigenvectors as its columns and $\mathbf{D}$ is a $(j-1) \times (j-1)$ diagonal matrix with sufficiently high constants on its diagonal, and search for its principal component. The matrix $\overline{\mathbf{K}}$ does not necessarily satisfy the sparse block structure of $\mathbf{K}$ so we cannot use the solution in Proposition 1 directly. Fortunately, it can be easily adjusted since the modification to $\mathbf{K}$ is of low rank.

Proposition 2: Let $\mathbf{K}$ be a symmetric matrix with $\mathbf{K}_{a,b} = \mathbf{K}_{b,a}^T = \mathbf{0}$. Then, the constraint

$$t < \mathrm{eig}_{\min} \left(\begin{bmatrix} \mathbf{K}_{a,a} & \mathbf{K}_{a,c} & \mathbf{0} \\ \mathbf{K}_{c,a} & \mathbf{K}_{c,c} & \mathbf{K}_{c,b} \\ \mathbf{0} & \mathbf{K}_{b,c} & \mathbf{K}_{b,b} \end{bmatrix} + \mathbf{U}\mathbf{D}\mathbf{U}^T \right) \tag{38}$$

is equivalent to the following pair of constraints

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

$$t \quad < \quad \mathrm{eig}_{\min} \left(\mathbf{K}_{C_1,C_1} + \left[\overline{\mathbf{U}} \right]_{C_1,:} \overline{\mathbf{D}} \left[\overline{\mathbf{U}} \right]_{C_1,:}^T \right) \tag{40}$$

where

$$\left[\overline{\mathbf{U}} \right]_{C_1,:} \quad = \quad \begin{bmatrix} \mathbf{0} & \\ & [\mathbf{U}]_{C_1} \\ \mathbf{I} & \end{bmatrix} \tag{41}$$

$$\overline{\mathbf{D}} \quad = \quad \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} - \mathbf{M_U}(t) \tag{42}$$

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

$$\mathbf{M_U}(t) \quad = \quad \begin{bmatrix} \mathbf{K}_{c,b} \\ \mathbf{D}[\mathbf{U}]_{b,:}^T \end{bmatrix} \left(\mathbf{K}_{b,b} + [\mathbf{U}]_{b,:} \mathbf{D}[\mathbf{U}]_{b,:}^T - t\mathbf{I} \right)^{-1} \begin{bmatrix} \mathbf{K}_{b,c} & [\mathbf{U}]_{b,:} \mathbf{D} \end{bmatrix}. \tag{43}$$

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

Thus, the solution to the j'th largest eigenvalue is similar to the method in Section II-C. The only difference is that the messages are slightly larger. Each message is a matrix of size $|S| + j - 1 \times |S| + j - 1$. In practice, dimensionality reduction involves only a few principal components and this method is efficient when $|S| + j - 1$ is considerably less than p (the size of the messages in a fully centralized protocol).

The higher order components can therefore be found in a distributed manner as detailed in Section II-D above.

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 graphs that can be recursively subdivided into the two cliques graph in Fig. 1. Therefore, this section consists of numerous technical definitions taken from [16] followed by a recursive application of the previous results.

An undirected graph $\mathcal{G}$ is a set of nodes connected by undirected edges. A random vector $\mathbf{x}$ satisfies the Markov property with respect to $\mathcal{G}$, if for any pair of non-adjacent nodes the corresponding pair of random variables are conditionally independent on the rest of the elements in $\mathbf{x}$. In the Gaussian distribution, this definition results in sparsity in the concentration domain. If $\mathbf{K}$ is the concentration matrix of a jointly Gaussian multivariate $\mathbf{x}$ that satisfies $\mathcal{G}$, then $[\mathbf{K}]_{i,j} = 0$ for any pair $\{i, j\}$ of non-adjacent nodes.

Decomposable graphs are a specific type of graph which possess an appealing structure. A graph is decomposable if it can be recursively be subdivided into disjoint sets of nodes a, b and c, where c separates a and b, and c is complete, i.e., there are no edges between a and b and all the nodes within c are connected by an edge. Clearly, the simplest non-trivial decomposable graph is the two cliques graph in Fig. 1.

A clique is a maximal subset of nodes which is fully connected. It is convenient to represent a decomposable graph using a sequence of cliques $C_1, \cdots, C_K$ 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

$$H_j = C_1 \cup C_2 \cup \cdots \cup C_j, \quad j = 1, \cdots, K \tag{44}$$

$$S_j = H_{j-1} \cap C_j, \quad j = 2, \cdots, K \tag{45}$$

$$R_j = H_j \backslash H_{j-1}, \quad j = 2, \cdots, K. \tag{46}$$

Note that this perfect elimination order induces an inherent asymmetry between the cliques which will be used in our recursive solution below. The two cliques graph in Fig. 1 is a simple example of a decomposable graph with $C_1 = \{a, c\}$, $C_2 = \{c, b\}$, $S_2 = \{c\}$, $H_1 = \{a, c\}$, $H_2 = \{a, c, b\}$ and $R_2 = \{b\}$. Accordingly, $S_2 = \{c\}$ separates $H_1 \backslash S_2 = \{a\}$ from $C_2 \backslash S_2 = \{b\}$.

Similarly to the previous section, global ML estimation of the concentration matrix in decomposable Gaussian graphical model has a simple closed form. It can be computed in a distributed manner:

$$\mathbf{K} = \sum_{k=1}^{K} \left[\tilde{\mathbf{K}}^{C_k, C_k} \right]^0 - \sum_{k=2}^{K} \left[\tilde{\mathbf{K}}^{S_k, S_k} \right]^0 \tag{47}$$

where the local estimates are defined as:

$$\tilde{\mathbf{K}}^{C_k,C_k} = \left(\tilde{\mathbf{S}}^{C_k,C_k}\right)^{-1}, \quad k = 1, \cdots, K \tag{48}$$

$$\tilde{\mathbf{K}}^{S_k,S_k} = \left(\tilde{\mathbf{S}}^{S_k,S_k}\right)^{-1}, \quad k = 2, \cdots, K, \tag{49}$$

and

$$\tilde{\mathbf{S}}^{C_k,C_k} = \frac{1}{n} \sum_{i=1}^{n} [\mathbf{x}_i]_{C_k} [\mathbf{x}_i]_{C_k}^T, \quad k = 1, \cdots, K \tag{50}$$

$$\tilde{\mathbf{S}}^{S_k,S_k} = \frac{1}{n} \sum_{i=1}^{n} [\mathbf{x}_i]_{S_k} [\mathbf{x}_i]_{S_k}^T, \quad k = 2, \cdots, K. \tag{51}$$

The zero fill-in operator $[\cdot]^0$ in (47) outputs a matrix of the same dimension as $\mathbf{K}$ where the argument occupies the appropriate sub-block and the rest of the matrix has zero valued elements (See (8) for a two clique example, and [16] for the exact definition of this operator).

DPCA can be recursively implemented by using the previous two clique solution. Indeed, Proposition 1 shows that the eigenvalue inequality

$$t < \mathrm{eig}_{\min}(\mathbf{K}) \tag{52}$$

is equivalent to two adjusted local eigenvalue inequalities

$$t \quad < \quad \mathrm{eig}_{\min}\left(\mathbf{K}'_{R_K}(t)\right) \tag{53}$$

$$t \quad < \quad \mathrm{eig}_{\min}\left(\mathbf{K}'_{H_{K-1}}(t)\right) \tag{54}$$

where

$$\mathbf{K}'_{R_K}(t) \quad = \quad \mathbf{K}_{R_K, R_K} \tag{55}$$

$$\mathbf{K}'_{H_{K-1}}(t) \quad = \quad \mathbf{K}_{H_{K-1}, H_{K-1}}(t) - [\mathbf{M}_k(t)]^0. \tag{56}$$

where $\mathbf{M}_k(t)$ is a message as in (23) and $[\cdot]^0$ is the zero fill-in operator. Next, we can apply Schur's Lemma again and replace (54) with two additional inequalities:

$$t \quad < \quad \mathrm{eig}_{\min}\left(\mathbf{K}'_{R_K}(t)\right) \tag{57}$$

$$t \quad < \quad \mathrm{eig}_{\min}\left(\mathbf{K}''_{R_{K-1}}(t)\right) \tag{58}$$

$$t \quad < \quad \mathrm{eig}_{\min}\left(\mathbf{K}''_{H_{K-2}}(t)\right) \tag{59}$$

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 with minimal message passing between the cliques, and any line-search can efficiently solve DPCA.

Specifically, in Algorithm 1 displayed below we provide a pseudo code for DPCA that solves for t using the bisection method. Given initial bounds

$$L \leq \mathrm{eig}_{\min}(\mathbf{K}) \leq U, \tag{60}$$

Algorithm 1 is guaranteed to find the minimal eigenvalue up to any required tolerance ϵ within $\log_2 \frac{U-L}{\epsilon}$ 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 $k = 2, \cdots, K$. A simple choice for the bounds is $L = 0$ since $\mathbf{K}$ is positive definite, and

$$U = \min_{k=1,\cdots,K} \{\mathrm{eig}_{\min}(\mathbf{K}_{\mathrm{C_k},\mathrm{C_k}})\} \tag{61}$$

as proved in the Appendix.

Given a principal eigenvalue λ, its corresponding eigenvector can be computed by solving $\mathbf{Qu} = \mathbf{0}$ where $\mathbf{Q} = \mathbf{K} - \lambda \mathbf{I}$ as detailed in Section II-D. Beginning with $k = K$ we partition H_k into R_k and H_{k-1} and 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 (29). A pseudo code of this method is provided in Algorithm 2 below.

Algorithm 1 can be easily extended to compute higher order eigenvalues through application of Proposition 2. For this purpose, note that the inequality in (40) has the same structure as (38) 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 2 finds the associated eigenvector in a distributed and efficient manner.

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., $C_1 =$

Algorithm 1: Bisection line search for DPCA

Input: $\mathbf{K}$, L, U, ϵ, 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 $t < \mathrm{eig}_{\min}\left(\mathbf{Q}_{R_k, R_k}\right)$ **then**

            $\mathbf{M}_k\left(t\right) = \mathbf{Q}_{S_k, R_k}\left(\mathbf{Q}_{R_k, R_k} - t\mathbf{I}\right)^{-1}\mathbf{Q}_{R_k, S_k}$

            $\mathbf{Q}_{S_k, S_k} = \mathbf{Q}_{S_k, S_k} - \mathbf{M}_k\left(t\right)$

        else

            $U = t$

            break loop

        end

    end

    if $U > t$ **then**

        if $t < \mathrm{eig}_{\min}\left(\mathbf{Q}_{C_1, C_1}\right)$ **then**

            $L = t$

        else

            $U = t$

        end

    end

end

$\{1, \cdots, 100, 301, \cdots, 305\}$, $C_2 = \{101, \cdots, 200, 301, \cdots, 305\}$, and $C_3 = \{201, \cdots, 300, 301, \cdots, 305\}$. 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 (47)-(51) 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. Figure 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

---

Algorithm 2: Eigenvector computation via $\mathbf{Qu} = \mathbf{0}$

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**

$\quad\quad \mathbf{M}_k = \mathbf{Q}_{S_k,R_k}\mathbf{Q}_{R_k,R_k}^{-1}\mathbf{Q}_{R_k,S_k}$

$\quad\quad \mathbf{Q}_{S_k,S_k} = \mathbf{Q}_{S_k,S_k} - \mathbf{M}_k$

$\quad\quad k = k - 1$

end

$\mathbf{u}\left(C_k\right) = \mathbf{u}_{\mathrm{null}}\left(\mathbf{Q}_{C_k,C_k}\right)$

for $k = k + 1, \cdots, K$ **do**

$\quad\quad \mathbf{u}\left(R_k\right) = -\mathbf{Q}_{R_k,R_k}^{-1}\mathbf{Q}_{R_k,S_k}\mathbf{u}\left(S_k\right)$

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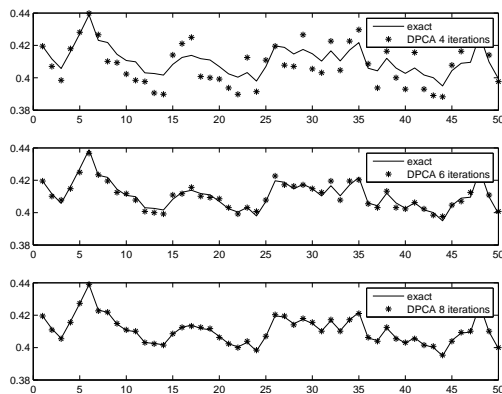
Fig. 2.   Iterations of the DPCA bisection line-search in a time varying scenario.

through two messages of size 5×5. For comparison, a centralized solution would require 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 IN NETWORKS

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 [15], [12], we consider a real world dataset of Abilene, the Internet2 backbone network. This network carries traffic from universities in the United States. Figure 3 shows its connectivity map consisting of 11 routers and 41 links (each edge corresponds to two links and there are additional links from each of the nodes to itself). Examining the network it is easy to see that the links on the east and west sides of the map are separated through six coupling links: DNVR-KSCY, SNVA-KSCY and LOSA-HSTN. Thus, our first approximated decomposable graph, denoted by $\mathcal{G}_{2\text{cliques}}$, consists of two cliques: an eastern clique and a western clique coupled by these six links. Graph $\mathcal{G}_{2\text{cliques}}$ corresponds to a decomposable concentration matrix with a sparsity level of 0.33. Our second decomposable graph denoted by $\mathcal{G}_{3\text{cliques}}$ is obtained by redividing the eastern clique again into two cliques separated through the four coupling links: IPLS-CHIN and ATLA-WASH. Its corresponding concentration matrix has a sparsity level of 0.43. Finally, for comparison we randomly generate an arbitrary graph $\mathcal{G}_{\text{random}}$ over the Abilene nodes, with an identical structure as $\mathcal{G}_{3\text{cliques}}$ (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 load 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 $\mathcal{G}_{2\text{cliques}}$, $\mathcal{G}_{3\text{cliques}}$ 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 model for Abilene is a good approximation and does not cause substantial loss of information (at least for the purpose of anomaly
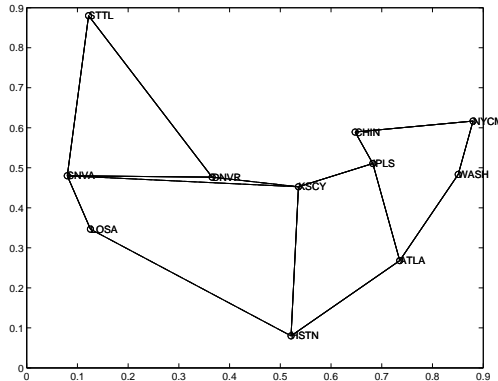
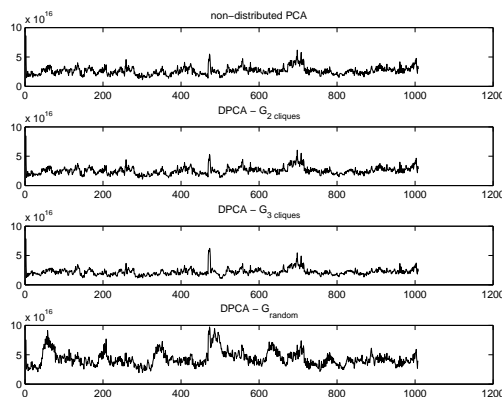Fig. 3.   Map of the Abilene network.

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

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 non-distributed PCA. These conclusions are supported 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 $\mathcal{G}_{2\text{cliques}}$ results in minimal error, $\mathcal{G}_{3\text{cliques}}$ provides a reasonable tradeoff between performance and computational complexity (through its increased sparsity level), while graph $\mathcal{G}_{\text{random}}$ 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.
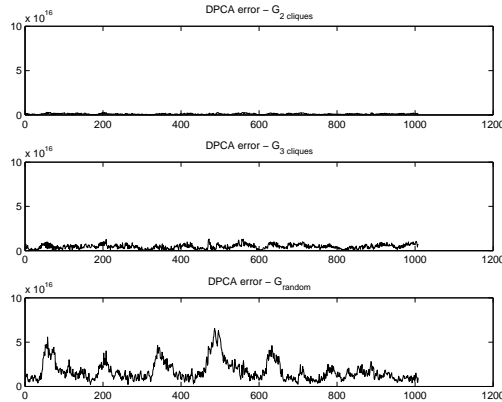
Fig. 5.   Absolute error in projection into anomaly subspace with different graphical models.

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 is also important to assess its sensitivity to non-Gaussian sources. Finally, alternative methods to ML in singular and ill conditioned scenarios should be considered.

VII. ACKNOWLEDGMENTS

APPENDIX

In this appendix, we prove that the minimal eigenvalue of a $p \times p$ symmetric matrix $\mathbf{K}$ is less than or equal to the minimal eigenvalue of any of its sub-matrices, say $\mathbf{K}_{a,a}$ for some set of indices a. For simplicity, we assume that $a = \{1, \cdots, p_a\}$ for some integer $p_a \leq p$. The proof is a simple application of the Rayleigh quotient characterization of the minimal eigenvalues:

$$\text{eig}_{\min}(\mathbf{K}) \;=\; \min_{\mathbf{u}} \frac{\mathbf{u}^T \mathbf{K} \mathbf{u}}{\mathbf{u}^T \mathbf{u}} \tag{62}$$

$$\leq \frac{\begin{bmatrix} \mathbf{v}^T & \mathbf{0}^T \end{bmatrix} \mathbf{K} \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix}}{\begin{bmatrix} \mathbf{v}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix}} \tag{63}$$

$$= \frac{\mathbf{v}^T \mathbf{K}_{a,a} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \tag{64}$$

$$= \min_{\mathbf{u}} \frac{\mathbf{u}^T \mathbf{K}_{a,a} \mathbf{u}}{\mathbf{u}^T \mathbf{u}} \tag{65}$$

$$= \mathrm{eig}_{\min}\{\mathbf{K}_{a,a}\} \tag{66}$$

where $\mathbf{v}$ is the optimal solution to (65).

REFERENCES

[1] T. W. Anderson. *An introduction to multivariate statistical analysis*. John Wiley and Sons, second edition edition, 1971.

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

[3] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, 9:485–516, March 2008.

[4] S. Boyd and L. Vandenberghe. *Introduction to Convex Optimization with Engineering Applications*. Stanford, 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 Processing Magazine*, 23(4):42– 55, July 2006.

[6] P. Chhabra, C. Scott, E. Kolaczyk, and M. Crovella. Distributed spatial anomaly detection. In *Proceedings of INFOCOM*, April 2008.

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

[8] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the LASSO. *Biostat*, 9(3):432 – 441, July 2008.

[9] M. Gastpar, P. L. Dragotti, and M. Vetterli. The distributed Karhunen Loeve transform. *IEEE Trans. on Information Theory*, 52(12):5177–5196, Dec. 2006.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins, 1983.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer, New York, 2001.

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

[13] M. I. Jordan. *Introduction to graphical models*. Unpublished, 2008.

[14] H. Kargupta, W. Huang, K. Sivakumar, and E. Hohnson. Distributed clustering using collective principal component analysis. *Knowledge and Information Systems*, 3(4):422–448, Nov. 2001.

[15] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. *SIGCOMM Comput. Commun. Rev.*, 34(4):219–230, 2004.

[16] S. L. Lauritzen. *Graphical models*, volume 17. Oxford Statistical Science Series, New York, 1996.

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

[18] O. Roy and M. Vetterli. Dimensionality reduction for distributed estimation in the infinite dimensional regime. *IEEE Trans. on Information Theory*, 54(2):1655–1669, April 2008.

[19] I. D. Schizas, G. B. Giannakis, and Z. Q. Luo. Distributed estimation using reduced-dimensionality sensor observations. *IEEE Trans. on Signal Processing*, 55(8):4284–4299, Aug. 2007.

[20] Y. Weiss and W. T. Freeman. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Comp.*, 13(10):2173–2200, 2001.

[21] J. J. Xiao, A. Ribeiro, Z. Q. Luo, and G. B. Giannakis. Distributed compression-estimation using wireless sensor networks. *Signal Processing Magazine*, 23(4):27– 41, July 2006.

[22] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):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. on Signal Processing*, 53(5):1631–1639, May 2005.