PRINCIPAL COMPONENT ANALYSIS IN DECOMPOSABLE GAUSSIAN GRAPHICAL MODELS

Ami Wiesel and Alfred O. Hero III

Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109, USA.

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.

Index Terms— Principal component analysis, graphical models, distributed data mining.

1. 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. Gaussian graphical models, also known as covariance selection models, provide a graphical representation of the conditional independence structure within the Gaussian distribution [1, 2]. This structure allows for efficient distributed implementation of statistical inference algorithms, e.g., the well known belief propagation method. 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 DPCA is decentralized PCA. Distributed dimensionality reduction plays a leading role in distributed estimation and compression theory in wireless sensor networks and decentralized data mining techniques, e.g., [3, 4, 5] and references within. It has also been used in anomaly detection in computer networks [6]. Graphical models have been recently developed for these setting within the statistics and signal processing communities, e.g., [7, 8].

DPCA exploits the structure of the decomposable graphical model in order to distribute the PCA computation. 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 eigenvalue problem using a sequence of local eigenvalue problems in each of the cliques of the 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 Internet traffic on the Abilene backbone network [9, 6]. 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 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 their information is approximately independent conditioned on the values of the nodes on the path. We examine the validity of this claim by applying DPCA in the context of anomaly detection in the Abilene network using a real-world dataset.

The outline of the paper is as follows. In Section 2, we begin with a brief review on decomposable graphical models. In Section 3 we introduce the problem formulation and solution to DPCA. Next, in Section 4 we illustrate its application to anomaly detection using a real-world dataset from the Abilene backbone network.

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, $\text{eig}_{\min}(\mathbf{X})$ is the minimum eigenvalue of square symmetric matrix $\mathbf{X}$, $\mathbf{u}_{\min}(\mathbf{X})$ is its associated eigenvector, $\text{eig}_{\max}(\mathbf{X})$ 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.

2. REVIEW ON DECOMPOSABLE GRAPHICAL MODELS

Graphical models are intuitive characterizations of conditional independence structures within multivariate distributions [1, 2]. 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 given the values of the rest of the elements in $\mathbf{x}$. In the Gaussian distribution, this definition has an appealing property. 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 graphs which pos-

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 grant No. FA9550-06-1-0324.

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

sess a special structure. A graph is decomposable (or chordal) 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 in c are connected by an edge. The simplest non-trivial decomposable graph is:

$$\overset{a}{\bigcirc}\longleftrightarrow\overset{c}{\bigcirc}\longleftrightarrow\overset{b}{\bigcirc} \tag{1}$$

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

$$\begin{aligned} H_j &= C_1 \cup C_2 \cup \cdots \cup C_j, \quad j = 1, \cdots, K \\ S_j &= H_{j-1} \cap C_j, \quad j = 2, \cdots, K \\ R_j &= H_j \backslash H_{j-1}, \quad j = 2, \cdots, K. \end{aligned} \tag{2}$$

For example, in the graph in (1) we define $C_1 = \{a, c\}$, $C_2 = \{c, b\}$, $S_2 = \{c\}$, $H_1 = \{a, c\}$, $H_2 = \{a, c, b\}$ and $R_2 = \{b\}$. Accordingly, $S_2 = \{c\}$ separates $H_1 \backslash S_2 = \{a\}$ from $C_2 \backslash S_2 = \{b\}$.

3. DECOMPOSABLE PRINCIPAL COMPONENT ANALYSIS

3.1. Problem Formulation

Let $\mathcal{G}$ be a decomposable graph with the clique structure described in (2) above. Let $\mathbf{x}$ be a zero mean, Gaussian random vector which follows the Markov properties of $\mathcal{G}$.

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 locally distributed in the sense that each clique has access only to $[\mathbf{x}_i]_{C_k}$ for $i = 1, \cdots, n$ and $k = 1, \cdots, K$. Using this data and message passing between the 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 $[\mathbf{u}]_{C_k}$.

3.2. Solution: covariance estimation

Similarly to PCA, the first phase of DPCA is covariance estimation. An appealing property of decomposable Gaussian graphical model is that global Maximum Likelihood (ML) estimation of the concentration matrix has a simple closed form which can be computed in a distributed manner [1]:

$$\mathbf{K} = \sum_{k=1}^{K} \left[\tilde{\mathbf{K}}^{C_k, C_k} \right]^0 - \sum_{k=2}^{K} \left[\tilde{\mathbf{K}}^{S_k, S_k} \right]^0 \tag{3}$$

where the local estimates are defined as the inverses of the local sample covariances:

$$\tilde{\mathbf{K}}^{C_k, C_k} = \left(\frac{1}{n} \sum_{i=1}^{n} [\mathbf{x}_i]_{C_k} [\mathbf{x}_i]_{C_k}^T \right)^{-1}, \quad k = 1, \cdots, K \tag{4}$$

$$\tilde{\mathbf{K}}^{S_k, S_k} = \left(\frac{1}{n} \sum_{i=1}^{n} [\mathbf{x}_i]_{S_k} [\mathbf{x}_i]_{S_k}^T \right)^{-1}, \quad k = 2, \cdots, K, \tag{5}$$

The zero fill-in operator $[\cdot]^0$ in (3) 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. For example, in the simple two cliques graph in (1) we obtain:

$$\mathbf{K} = \begin{bmatrix} \tilde{\mathbf{K}}^{C_1, C_1} & & \mathbf{0} \\ & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}^{C_2, C_2} & \\ \mathbf{0} & & \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}^{S, S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{6}$$

where we have used the order $\mathbf{x}^T = [\, a \;\; c \;\; b \,]$. Note the sparse block structure of the ML estimate in which $\mathbf{K}_{a,b} = \mathbf{K}_{b,a}^T = \mathbf{0}$. On the other hand, the global ML covariance estimate is the inverse of the concentration estimate, i.e., $\mathbf{S} = \mathbf{K}^{-1}$, and is not necessarily a sparse matrix.

3.3. Solution: principal eigenvalue

The next phase in DPCA is principal eigenvalue computation. Given the global estimate $\mathbf{S}$, the PCA objective function is estimated as $\mathbf{u}^T \mathbf{S} \mathbf{u}$ and maximized subject to a norm constraint to yield

$$\mathrm{eig}_{\max}(\mathbf{S}) = \left\{ \begin{array}{ll} \max_{\mathbf{u}} & \mathbf{u}^T \mathbf{S} \mathbf{u} \\ \text{s.t.} & \mathbf{u}^T \mathbf{u} = 1. \end{array} \right. \tag{7}$$

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

$$\mathrm{eig}_{\max}(\mathbf{S}) = \frac{1}{\mathrm{eig}_{\min}(\mathbf{K})}, \tag{8}$$

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 (6).

We now return to the problem of finding $\lambda = \mathrm{eig}_{\min}(\mathbf{K})$ in a distributed manner. We begin by expressing λ as a trivial line-search problem:

$$\lambda = \sup \quad t \quad \text{s.t.} \quad t < \mathrm{eig}_{\min}(\mathbf{K}) \tag{9}$$

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, we will now show that checking the feasibility of a given t can be done in a distributed manner. For simplicity, we begin with the simple two cliques graph in (1), and obtain the following proposition.

Proposition 1 *Let* $\mathbf{K}$ *be a symmetric matrix with* $\mathbf{K}_{a,b} = \mathbf{K}_{b,a}^T = \mathbf{0}$*. Then, the constraint*

$$t < eig_{\min}\left(\begin{bmatrix} \mathbf{K}_{a,a} & \mathbf{K}_{a,c} & \mathbf{0} \\ \mathbf{K}_{c,a} & \mathbf{K}_{c,c} & \mathbf{K}_{c,b} \\ \mathbf{0} & \mathbf{K}_{b,c} & \mathbf{K}_{b,b} \end{bmatrix} \right) \tag{10}$$

is equivalent to the following pair of constraints

$$
\begin{aligned}
t &< \quad eig_{\min}\left(\mathbf{K}_{R_2,R_2}\right) && (11)\\
t &< \quad eig_{\min}\left(\mathbf{K}_{H_1,H_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(t\right) \end{bmatrix}\right) && (12)
\end{aligned}
$$

with the message matrix *defined as*

$$\mathbf{M}\left(t\right) = \mathbf{K}_{S_2,R_2}\left(\mathbf{K}_{R_2,R_2} - t\mathbf{I}\right)^{-1}\mathbf{K}_{R_2,S_2}. \qquad (13)$$

The proof is obtained by rewriting (10) as a linear matrix inequality and using Schur's Lemma [10, Appendix A5.5]. More details can be found in [11].

Proposition 1 shows that the global eigenvalue inequality is equivalent to two local eigenvalue inequalities. Using the recursive structure of decomposable graphs, we can recursively decouple these inequalities again until we obtain one local eigenvalue problem in each clique. Thus, for any given t we can check the feasibility by solving local eigenvalue problems and message passing via $\mathbf{M}\left(t\right)$ whose dimensions are equal to the cardinalities of the separators. The optimal global eigenvalue is then defined as the maximal globally feasible t.

In Algorithm 1 displayed below we provide a pseudo code for DPCA that solves for t using the bisection method. Given initial bounds $L \leq \text{eig}_{\min}\left(\mathbf{K}\right) \leq U$, 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 $\mathbf{M}_k\left(t\right)$ whose dimensions are equal to the cardinalities of S_k for $k = 2, \cdots, K$. A simple choice for the bounds is $L = 0$ and $U = \min_{k=1,\cdots,K}\{\text{eig}_{\min}\left(\mathbf{K}_{C_k,C_k}\right)\}$.

Algorithm 1: Bisection line search for DPCA

```
Input: $\mathbf{K}$, $L$, $U$, $\epsilon$, clique tree structure
Output: $t$
while $U - L > \epsilon$ do
    $t = \left(U + L\right)/2$
    $\mathbf{Q} = \mathbf{K}$
    for $k = K, \cdots, 2$ do
        if $t < \text{eig}_{\min}\left(\mathbf{Q}_{R_k,R_k}\right)$ 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 $t < \text{eig}_{\min}\left(\mathbf{Q}_{C_1,C_1}\right)$ then
            $L = t$
        else
            $U = t$
        end
    end
end
```

3.4. Solution: principal eigenvector

After we obtain the minimal eigenvalue λ, we can easily recover its corresponding eigenvector using similar local computations. For simplicity, we begin again with the two cliques case in (6). Usually, we have the strict inequality

$$\text{eig}_{\min}\left(\mathbf{K}_{R_2,R_2}\right) > \text{eig}_{\min}\left(\mathbf{K}_{H_1,H_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(\lambda\right) \end{bmatrix}\right), \qquad (14)$$

in which case λ is given by the right hand side of (14) and the corresponding eigenvector $\mathbf{u}$ is [11]

$$
\begin{aligned}
\left[\mathbf{u}\right]_{C_1} &= \mathbf{u}_{\min}\left(\mathbf{K}_{H_1,H_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(\lambda\right) \end{bmatrix}\right) && (15)\\
\left[\mathbf{u}\right]_{b} &= -\left[\mathbf{K}_{R_2,R_2} - \lambda\mathbf{I}\right]^{-1}\mathbf{K}_{R_2,S_2}\left[\mathbf{u}\right]_c. && (16)
\end{aligned}
$$

In general, it is possible that the inequality in (14) will not hold. In this case, λ is equal to the left hand side of (14), and the corresponding eigenvector is simply

$$
\begin{aligned}
\left[\mathbf{u}\right]_{C_1} &= \mathbf{0} && (17)\\
\left[\mathbf{u}\right]_{R_2} &= \mathbf{u}_{\min}\left(\mathbf{K}_{R_2,R_2}\right). && (18)
\end{aligned}
$$

This singular case is highly unlikely as the probability of (17) in continuous models is zero. However, for finite register length computations this condition needs to be checked.

The generalization of the two cliques solution to decomposable graphs is a straightforward application of their recursive definition. A pseudo code of this method is provided in Algorithm 2 below. The first **while** loop searches backwards for the clique associated with λ, whereas the second **for** loop propagates the solution forward.

Algorithm 2: Eigenvector computation

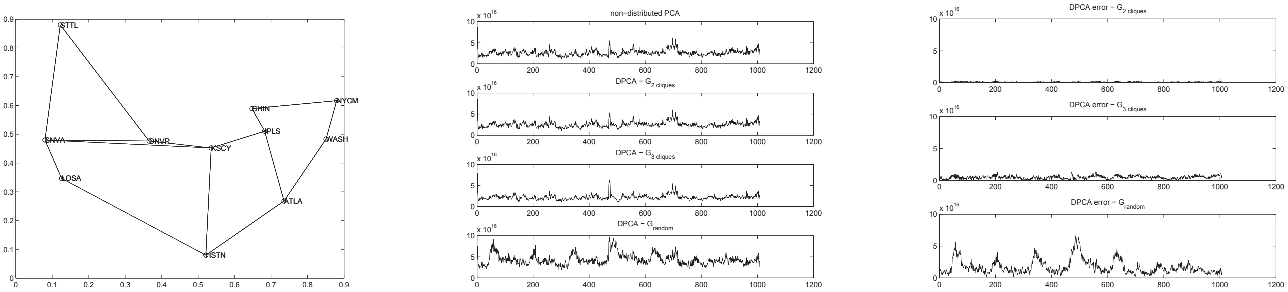
```
Input: $\mathbf{K}$, $\lambda$, clique tree structure
Output: $\mathbf{u}$
$\mathbf{u} = \mathbf{0}$
$\mathbf{Q} = \mathbf{K}$
$k = K$
while $(k > 1)$ & $(\lambda < \text{eig}_{\min}\left(\mathbf{Q}_{R_k,R_k}\right))$ do
    $\mathbf{M}_k\left(\lambda\right) = \mathbf{Q}_{S_k,R_k}\left(\mathbf{Q}_{R_k,R_k} - \lambda\mathbf{I}\right)^{-1}\mathbf{Q}_{R_k,S_k}$
    $\mathbf{Q}_{S_k,S_k} = \mathbf{Q}_{S_k,S_k} - \mathbf{M}_k\left(\lambda\right)$
    $k = k - 1$
end
$\mathbf{u}\left(C_k\right) = \mathbf{u}_{\min}\left(\mathbf{Q}_{C_k,C_k}\right)$
for $k = k + 1, \cdots, K$ do
    $\mathbf{u}\left(R_k\right) = -\left(\mathbf{Q}_{R_k,R_k} - \lambda\mathbf{I}\right)^{-1}\mathbf{Q}_{R_k,S_k}\mathbf{u}\left(S_k\right)$
end
```

3.5. Solution: higher order components

Algorithms 1 and 2 can be extended to compute higher order components. We begin with the first component $\mathbf{u}_1$, find it and then rerun the algorithms with respect to $\mathbf{K} + \mu\mathbf{u}_1\mathbf{u}_1^T$ where μ is a sufficiently large number which ensures that $\mathbf{u}_1$ will not be found again. This extension is not straightforward as $\mathbf{K} + \mu\mathbf{u}_1\mathbf{u}_1^T$ is no longer a sparse matrix. Nonetheless, the modification is of low rank and can be handled efficiently with only a slight increase in the required message passing. More details about this procedure are available in [11].

4. 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



(a) Map of the network (b) Projection into anomaly subspace (c) Absolute error in projection

Fig. 1. Anomaly detection in the Abilene backbone network using DPCA.

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 [9, 6], we consider a real world dataset of Abilene, the Internet2 backbone network. This network carries traffic between universities in the United States. Figure 1(a) shows the router 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 the period 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. 1(b). 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. 1(b)., 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 detection). On the other hand, the residual norm using the random graph model is a poor approximation as it does not preserve the anomalies detected by the full non-distributed PCA. These conclusions are supported in Fig. 1(c) 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.

5. REFERENCES

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

[2] A. P. Dempster, “Covariance selection,” *Biometrics*, vol. 28, pp. 157–175, 1972.

[3] M. Gastpar, P. L. Dragotti, and M. Vetterli, “The distributed Karhunen Loeve transform,” *IEEE Trans. on Information Theory*, vol. 52, no. 12, pp. 5177–5196, Dec. 2006.

[4] J. J. Xiao, A. Ribeiro, Z. Q. Luo, and G. B. Giannakis, “Distributed compression-estimation using wireless sensor networks,” *Signal Processing Magazine*, vol. 23, no. 4, pp. 27–41, July 2006.

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

[6] 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.

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

[8] 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*, vol. 23, no. 4, pp. 42– 55, July 2006.

[9] Anukool Lakhina, Mark Crovella, and Christophe Diot, “Diagnosing network-wide traffic anomalies,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 219–230, 2004.

[10] S. Boyd and L. Vandenberghe, *Introduction to Convex Optimization with Engineering Applications*, Stanford, 2003.

[11] A. Wiesel and A. O. Hero III, “Decomposable principal component analysis,” *Submitted to IEEE Trans. on Signal Processing, Aug. 2008, available in arXiv 0808.2337*.