Space-time signal processing for distributed pattern detection in sensor networks

*Randy Paffenroth, Philip du Toit, Ryan Nong, Louis Scharf, *Life Fellow, IEEE,*
Anura Jayasumana, *Senior Member, IEEE,* and Vidarshana Bandara

***Abstract*—A theory and algorithm for detecting and classifying weak, distributed patterns in network data is presented. The patterns we consider are anomalous temporal correlations between signals recorded at sensor nodes in a network. We use robust matrix completion and second order analysis to detect distributed patterns that are not discernible at the level of individual sensors. When viewed independently, the data at each node cannot provide a definitive determination of the underlying pattern, but when fused with data from across the network the relevant patterns emerge. We are specifically interested in detecting weak patterns in computer networks where the nodes (terminals, routers, servers, *etc.*) are sensors that provide measurements (of packet rates, user activity, central processing unit usage, *etc.*). The approach is applicable to many other types of sensor networks including wireless networks, mobile sensor networks, and social networks where correlated phenomena are of interest.**

***Index Terms*—Pattern detection, matrix completion, robust principal component analysis, anomaly detection, ℓ_1 methods.**

I. Introduction

We present an approach for detecting sparsely correlated phenomena in sensor networks. The patterns of interest are *weak* and *distributed across the network* so that they are inherently undetectable at the level of an individual node. Only when data is aggregated across the network both *spatially* and *temporally*, and appropriately analyzed are the relevant patterns revealed — an approach we refer to as *space-time signal processing*.

During a distributed attack, the time series at any given link or node may not appear suspicious or unusual, but when examined in the context of multiple correlated links with respect to prevailing network conditions, the distributed pattern appears as a discernible correlated anomaly.

Our work leverages recent progress in *compressed sensing* [1], *matrix completion* [2], *robust principal component analysis* [3]–[5], and *simple model discovery* [6] to provide a mathematical foundation and computational framework for an *unsupervised* learning algorithm that detects weak, distributed anomalies in sensor data. This is in contrast to supervised learning algorithms in data mining [7]–[9] and machine learning [10]–[13] that typically require large amounts of labeled training data, predefined models, or expert knowledge.

Our definition of "pattern" is based upon the principles of *low-rank* and *sparsity*. Data collected from a network is flagged as *patterned* if a sparse subset of nodes exhibits correlations that cannot be explained by the low-rank background correlation that broadly affects the entire network. These correlated anomalies can range from short duration high intensity effects, to longer duration lower intensity effects. As demonstrated by our results on measured data from the Abilene Internet2 network, both types of phenomena appear in real data and can be detected by our second-order analysis.

We must clarify that detection of distributed patterns does not in and of itself constitute detection of *attacks* — anomalies do not necessarily imply malicious behavior. By design, our goal is not to assign value judgements to network behavior in order to determine whether actors are bad, nor do we provide a long list of semantic rules for identifying malicious intent. Rather, we sift through large amounts of raw data and pinpoint sensors displaying unusual patterns of behavior that are out of step with the norm.

Automated real-time detection of abnormally correlated conditions can in turn trigger efforts to mitigate attacks, and can invoke network management responses to better diagnose network problems and meet quality of service targets. Fast identification of geographically distributed sensors related to the same abnormal condition can lead to more focused and effective counter-measures.

The proposed algorithm helps to mitigate false alarms by only flagging patterns that are anomalously correlated across multiple sensors so that greater consensus is required before a flag is raised. Our approach is similar in spirit to recent work in *collaborative* anomaly detection [14], but our methods proceed from a more mathematical, rather than empirical, perspective.

The computational core of the analysis is phrased as a convex optimization problem that can be solved efficiently on large datasets. The algorithm can detect anomalies even when the data of interest is distributed across multiple databases and is too large to aggregate at a centralized location for processing. Recent techniques in matrix completion [2]–[4], [15] allow for efficient analysis of large correlation matrices that, due to constraints imposed by network topologies, may be only *partially observed*.

We anticipate that the mathematical framework and algorithms developed herein will be applicable to very general classes of sensor networks including networked infrastruc-

R. Paffenroth (corresponding author at randy.paffenroth@numerica.us), P. du Toit, and Ryan Nong are with Numerica Corporation, 4850 Hahns Peak, Suite 200, Loveland, CO 80538, USA.

L. Scharf is with the Department of Mathematics at Colorado State University, Fort Collins, CO 80523, USA.

A. Jayasumana, and V. Bandara are with the Department of Computer and Electrical Engineering at Colorado State University, Fort Collins, CO 80523, USA.

Manuscript received August 7, 2012.

tures, electrical grids, computer networks, aerial surveillance networks, disease outbreaks, and social networks. We focus on computer networks as a motivating application area; however, we do not make any assumptions or heuristic arguments that are specific to computer networks. The results and exposition in this paper extend the results reported in two conference papers presented at the 2012 SPIE meeting [16], [17].

Our contributions include a new formulation of Robust Principal Component Analysis that combines robustness to noise with partial observations. Our methods uses point-wise error constraints that allow entries of the matrix to have different noise properties, as opposed to the standard Frobenius norm approach that applies a single global noise constraint. We present a stability theorem and an algorithm for addressing noisy problems with partial observations based upon a novel equivalent problem formulation that allows solution of the optimization using a standard Alternating Direction Method of Multipliers. We apply the method to second-order matrices to detect sparsely correlated phenomena in measured data from the Abilene Internet2 network.

The remainder of the paper is organized as follows: Section II elaborates the underlying theory, assumptions, and implementation of the algorithm. Results and a discussion on the outcomes are presented in Section III. Concluding remarks along with future directions are highlighted in Section IV. The appendix provides proof of Theorem 2.1.

II. Methods, Assumptions, and Procedures

A. Theoretical Background for Matrix Decomposition

We use several matrix norms in our analysis. Let $\|\cdot\|_*$ denote the nuclear norm: if $\{\sigma_1, \cdots, \sigma_m\}$ are the singular values of matrix A, then

$$\|A\|_* := \sum_{i=1}^{m} \sigma_i \,.$$

The two-norm, denoted $\|\cdot\|$, returns the largest singular value of the operator A:

$$\|A\| := \max_i \sigma_i \,.$$

We also refer to the following elementwise norms: the Frobenius norm,

$$\|A\|_F := \sqrt{\sum_{ij} A_{ij}^2} \,,$$

the one-norm,

$$\|A\|_1 := \sum_{ij} |A_{ij}| \,,$$

and the inifinity-norm,

$$\|A\|_\infty := \max_{ij} |A_{ij}| \,.$$

Let $\langle \cdot, \cdot \rangle$ denote the matrix inner-product,

$$\langle A, B \rangle := \text{trace}(A^T B),$$

so that $\|A\|_F^2 = \langle A, A \rangle$. We introduce $\mathcal{P}_\Omega(A)$, the projection of the matrix A onto the set of entries indexed by the indices in the set Ω, as follows:

$$[\mathcal{P}_\Omega(A)]_{ij} := \begin{cases} A_{ij} & ij \in \Omega \\ 0 & \text{otherwise} \,. \end{cases} \tag{1}$$

Also define the *shrinkage* operator $\mathcal{S}_\epsilon : \mathbb{R} \to \mathbb{R}$ as:

$$\mathcal{S}_\epsilon(x) := \text{sign}(x) \max(|x| - \epsilon, 0). \tag{2}$$

This shrinkage operator can be extended to *matrix shrinkage* by applying the scalar shrinkage operator to each element of the matrix using a matrix-valued ϵ. Also, we define the *rank shrinkage* operator $\mathcal{D}_\tau : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ as the shrinkage operator on singular values:

$$\mathcal{D}_\tau(X) := U \mathcal{S}_\tau(\Sigma) V^*,$$

where $X = U \Sigma V^*$ is any singular value decomposition of X. We use the "$(\cdot)_0$" subscript to denote truth, as in the true L_0 and S_0, and $\hat{L}$ and $\hat{S}$ to denote quantities we recover from the given data using our algorithm.

In the prototypical Principal Component Pursuit (PCP) problem, we are given a matrix M that is formed by

$$M = L_0 + S_0,$$

where L_0 is low-rank, S_0 is sparse, and we are asked to recover L_0 and S_0. As noted in [4], the matrix decomposition problem may appear somewhat daunting. Given a full-rank matrix M, we must tease out the underlying low rank matrix L_0, and identify the sparse anomalies introduced by S_0, *without knowing a priori the true rank of L_0, and without knowing the number or locations of the nonzero entries in S_0*. Furthermore, the magnitude of the few nonzero entries in S_0 may be of arbitrarily large size. These difficulties may be further compounded by failure to observe a subset of entries in M, and by noise that adds small errors to each of the remaining entries.

Remarkably, Theorem 1.2 in [4] provides the following guarantee for matrix decomposition and recovery. If $M = L_0 + S_0$, and we are given only $\mathcal{P}_\Omega(M)$, and if certain identifiability, rank, and sparsity conditions on L_0, S_0, and Ω are met; then, with high probability the convex program:

[Principal Component Pursuit]

$$\begin{aligned} &\min_{L,S} \|L\|_* + \lambda \|S\|_1 \\ &\text{subject to} \quad \mathcal{P}_\Omega(M) = \mathcal{P}_\Omega(L + S), \end{aligned} \tag{3}$$

with $\lambda = \sqrt{\frac{m}{|\Omega|}}$ exactly recovers the low rank matrix L_0, as well as the entries of the sparse matrix $S_0' := \mathcal{P}_\Omega(S_0)$.[1] Thus, the use of the nuclear and ℓ_1 norms allows for the decomposition of a patterned correlation matrix into low-rank and sparse constituents using a convex program even when the correlation matrix is only *partially observed*. The problem

[1]Technical details of the theorem place conditions on the rank of L_0, the sparsity of S_0, and the size of the observed set Ω, and also requires that the columns of L_0 are *incoherent* — meaning far from the standard basis, and that the nonzero entries in S_0 are distributed uniformly in S.

of minimizing rank and sparsity subject to constraints is NP-hard. Relaxation from rank minimization to nuclear norm minimization, and from sparsity minimization to ℓ_1-norm minimization as indicated in the objective in (3) results in a convex optimization problem that can be efficiently solved, and that recovers the exact low-rank and sparse solution with high probability.

With regard to matrix decomposition, the literature also addresses the question of *stability*. Is PCP for performing matrix decomposition into low-rank and sparse components stable to the addition of small but *dense* noise? To that end, we are interested in recovering L_0 and S_0 from

$$M = L_0 + S_0 + Z_0, \tag{4}$$

where Z_0 is a dense matrix of small noise terms. In this case, the convex program of interest is:

[Principal Component Pursuit with Frobenius Constraint]

$$\begin{aligned} \min_{L,S} \ & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} \ & \|M - L - S\|_F \leq \delta \,. \end{aligned} \tag{5}$$

We are aided by the central result of [18] that shows that the error in the recovery of L_0 and S_0 in the presence of noise is bounded by the size of the noise, $\delta := \|Z_0\|_F$.

B. Extensions of Matrix Decomposition

Our program for detecting patterns in network data will require extending algorithms for PCP to include point-wise error constraints and partial observations. Let ϵ be the matrix of entrywise error bounds. Then, we propose to extend PCP by solving:

[Principal Component Pursuit with Entry-wise Constraints]

$$\begin{aligned} \min_{L,S} \ & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} \ & |\mathcal{P}_\Omega(M) - \mathcal{P}_\Omega(L+S)| \preceq \epsilon \end{aligned} \tag{6}$$

where the inequality constraint is enforced element wise. Stability of this program is guaranteed by the following theorem that requires incoherence conditions on L_0 detailed in Definition A.1 of Appendix A.

Theorem 2.1 (Modification of Theorem 2 in [5]): Suppose that $\mathcal{P}_\Omega(M) = \mathcal{P}_\Omega(L_0 + S_0 + Z_0)$ in which L_0 is $n \times n$ and obeys the incoherence conditions defined in Definition A.1 with parameter μ, and that the support of S_0 and Ω are both uniformly distributed. Then, if L_0, S_0', and Ω satisfy

$$\begin{aligned} \text{rank}(L_0) \leq \frac{\rho_r n}{\mu \log^2 n} \qquad & \text{and} \\ |\text{support}(S_0')| + |\Omega^c| \leq \rho_s n^2, & \end{aligned}$$

where Ω^c is the set of unobserved entries in M, and ρ_r and ρ_s are some sufficiently small positive constants; with high probability, for any Z_0 with $|Z_0| \preceq \epsilon$, the solution $(\hat{L}, \hat{S})$ to the convex program (6) with $\lambda = 1/\sqrt{n}$ and $\|\epsilon\|_1 = \delta$ satisfies,

$$\|\hat{L} - L_0\|_F^2 + \|\hat{S} - S_0'\|_F^2 \leq C n^2 \delta^2 \,,$$

where C is a numerical constant.

Proof: Proof is provided in Appendix A. ∎

To be sure, we observe that since $\|Z_0\|_F \leq \|Z_0\|_1$, the error bound obtained in [5] for the Frobenius norm constraint allows for larger noise matrices than can be obtained using point wise constraints.

C. Equivalent Formulation

Our pattern detection framework requires efficiently solving the convex program (6). Algorithms for solving the matrix decomposition problem for $M_0 = L_0 + S_0$ have been presented in [5] and [15]; however, to our knowledge, no algorithms have been explicitly presented for dealing with the case of matrix decomposition with *partial observations* and *entry-wise inequality constraints*. We have extended existing algorithms to efficiently deal with these cases. Our method of choice is the Augmented Lagrange Multiplier (ALM) method that provides an iterative procedure for updating both the current estimate for the optimal solution, and a Lagrange multiplier that enforces the constraints. As has been noted in [6], the iterative Augmented Lagrange Multiplier approach is significantly faster and less memory intensive than second-order semi-definite program methods.

There is an important issue that needs to be addressed before we can apply the ALM method. At each iteration, the ALM method requires minimization of a Lagrangian with respect to the decision variables L and S. Moreover, the Alternating Direction Method of Multipliers (ADMM) requires minimizing the Lagrangian first with respect to L (with S held fixed) and then with respect to S (with L held fixed) [19]. The issue is that the Lagrangian associated with (6) does not allow for a closed form optimization with respect to L as is required for efficient convergence of the algorithm. However, we now demonstrate how this obstacle can be overcome by introducing a convex program that is mathematically equivalent to (6) but still allows for direct and efficient application of the ALM method.

Let $\tilde{\epsilon}$ be defined as

$$\tilde{\epsilon}_{ij} = \begin{cases} \epsilon_{ij} & ij \in \Omega \\ \infty & \text{otherwise} \,. \end{cases} \tag{7}$$

Theorem 2.2 (Equivalent Convex Optimization): Consider the following optimization problem:

$$\begin{aligned} \min_{L,S} \ & \|L\|_* + \lambda \|\mathcal{S}_{\tilde{\epsilon}}(S)\|_1 \\ \text{subject to} \ & M = L + S \,. \end{aligned} \tag{8}$$

Provided that each of (6) and (8) has a unique minimizer, these two convex optimization problems are equivalent.

The proof of Theorem 2.2 rests largely on the following lemma:

Lemma 2.3: Consider the following two optimization problems

$$\begin{aligned} \min_{L,S} \ & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} \ & |M - L - S| \preceq \epsilon \,, \end{aligned} \tag{9}$$

and

$$\begin{aligned} &\min_{L,S} \|L\|_* + \lambda \|\mathcal{S}_\epsilon(S)\|_1 \qquad (10)\\ \text{subject to} \quad &M = L + S\,. \end{aligned}$$

Provided that each of (9) and (10) has its own unique minimizer, these two convex optimization problems are equivalent.

Proof: We prove this lemma by way of contradiction. Let

$$\begin{aligned} (L_1, S_1) = \arg\min_{L,S} &\|L\|_* + \lambda \|S\|_1 \qquad (11)\\ \text{subject to} \quad &|M - L - S| \preceq \epsilon, \end{aligned}$$

and

$$\begin{aligned} (L_2, S_2) = \arg\min_{L,S} &\|L\|_* + \lambda \|\mathcal{S}_\epsilon(S)\|_1 \qquad (12)\\ \text{subject to} \quad &M = L + S \end{aligned}$$

be the unique solutions to the two optimization problems (9) and (10), respectively.

Now, consider the following three cases:

Case 1: $\|L_1\|_* + \lambda \|S_1\|_1 < \|L_2\|_* + \lambda \|\mathcal{S}_\epsilon(S_2)\|_1$.

Consider the constrained optimization (9). Let $L^* := L_1$ and $S^* := M - L_1$. In light of the constraint in (9), it is readily observed that

$$|S_*| \preceq |S_1| + \epsilon.$$

Therefore, in addition to $M = L_* + S_*$, we have

$$\begin{aligned} \|L^*\|_* + \lambda \|\mathcal{S}_\epsilon(S^*)\|_1 &\leq \|L_1\|_* + \lambda \|S_1\|_1 \\ &< \|L_2\|_* + \lambda \|\mathcal{S}_\epsilon(S_2)\|_1, \end{aligned}$$

which contradicts (12).

Case 2: $\|L_2\|_* + \lambda \|\mathcal{S}_\epsilon(S_2)\|_1 < \|L_1\|_* + \lambda \|S_1\|_1$.

Consider the constrained optimization (10). Let $L^* := L_2$ and $S^* := \mathcal{S}_\epsilon(S_2)$. Then in addition to $|M - L^* - S^*| \preceq \epsilon$, we have

$$\|L^*\|_* + \lambda \|S^*\|_1 < \|L_1\|_* + \lambda \|S_1\|_1,$$

which contradicts (11).

Case 3: $\|L_1\|_* + \lambda \|S_1\|_1 = \|L_2\|_* + \lambda \|\mathcal{S}_\epsilon(S_2)\|_1$.

If $L_1 \neq L_2$ and $S_1 \neq \mathcal{S}_\epsilon(S_2)$, then as shown in 1. and 2., we can construct new solutions to both of the optimization problems that contradict the assumption that each of these two problems has a unique solution respectively. ∎

Now we provide a proof of Theorem 2.2 regarding optimization equivalence:

Proof: Consider the following optimization problem:

$$\begin{aligned} &\min_{L,S} \|L\|_* + \lambda \|S\|_1 \qquad (13)\\ \text{subject to} \quad &|M - L - S| \preceq \tilde{\epsilon}\,, \end{aligned}$$

where $\tilde{\epsilon}$ is defined in (7). With $\mathcal{P}_\Omega(A)$ defined in (1), a proof by way of contradiction similar to that of Lemma 2.3 shows that (6) and (13) are equivalent provided that (13) has a unique minimizer. Now an application of Lemma 2.3 shows that (13) and (8) are equivalent. ∎

By comparing (6) with (8), it should be observed that the shrinkage operator has moved from the constraint to the objective. Also note how the introduction of $\tilde{\epsilon}$ accounts for partial observations on the set Ω. This new formulation in (8) results in a modified Lagrangian:

$$\mathcal{L}(L, S, Y, \mu) := \|L\|_* + \lambda \|\mathcal{S}_{\tilde{\epsilon}}(S)\|_1 + \langle Y, H \rangle + \frac{\mu}{2} \langle H, H \rangle\,, \qquad (14)$$

where $H := M - L - S$ encodes the equality constraint. The key difference is that this new Lagrangian allows for minimization with respect to both L and S in closed form so that the ADMM can proceed efficiently.

D. Principal Component Pursuit (PCP) with Noise

Having motivated the reformulation of the convex program (6) into (8), we now provide detailed explanation of the algorithm for solving this latter formulation.

We refer to our algorithm for solving PCP in noisy environments using inequality constraints as eRPCA. In this acronym, the RPCA stands for "Robust Principal Component Analysis", while the "e" in eRPCA is a reminder that inequality constraints are enforced point-wise with matrix ϵ.

Each iteration of the eRPCA algorithm requires an optimization of the Lagrangian with respect to both decision variables L and S. Using the structure of the subgradients for the $\|\cdot\|_1$, and $\|\cdot\|_*$ norms, we perform this inner-loop optimization *analytically*, so that the overall optimization proceeds very quickly.

An outline for the eRPCA algorithm is provided in Algorithm 1. In Step 1, the optimal value of L (as provided by [20]) is

$$L = \mathcal{D}_{\mu^{-1}}(M - S + \mu^{-1} Y)\,.$$

All that remains is to describe the function `Find_Optimal_S`(M, L, ϵ, Y, μ) listed in the algorithm that returns the value of S that minimizes the Lagrangian. This will require minimizing a function of the form, $F : \mathbb{R}^{m \times m} \to \mathbb{R}$, defined by

$$\begin{aligned} F(S) := \frac{\lambda}{\mu} \|\mathcal{S}_\epsilon(S)\|_1 - \text{tr}\left(\frac{Y^T}{\mu}(S - (M - L))\right) \\ + \frac{1}{2}\text{tr}\left((S - (M - L))^T (S - (M - L))\right)\,. \end{aligned}$$

Simplifying substitutions may be made by defining $\alpha := \frac{\lambda}{\mu} > 0$, $\beta_{ij} := -\frac{1}{\mu} Y_{ij}$, and $\gamma_{ij} := M_{ij} - L_{ij}$. Consequently, the expression for $F(S)$ can be written as:

$$F(S) = \sum_{ij} \left[\alpha |\mathcal{S}_{\epsilon_{ij}}(S_{ij})| + \beta_{ij}(S_{ij} - \gamma_{ij}) + \frac{1}{2}\left[(S_{ij} - \gamma_{ij})\right]^2\right].$$

Importantly, for each index ij, we have a grouping of three terms that depend only on the entry S_{ij}, and there is no

Outline for the eRPCA Algorithm:
The following constants are provided in the problem data:

- the raw data matrix, $M' := \mathcal{P}_\Omega(M) \in \mathbb{R}^{m\times m}$,
- the matrix of point wise error bounds, $\tilde{\epsilon} \in \mathbb{R}^{m\times m}$,
- the scalar weighting factor, $\lambda \in \mathbb{R}$,

The algorithm will use the following internal variables:

- $Y \in \mathbb{R}^{m\times m}$,
- $L \in \mathbb{R}^{m\times n}$,
- $S \in \mathbb{R}^{m\times m}$,
- $\mu \in \mathbb{R}$, $\rho \in \mathbb{R}$,
- $converged \in \{\text{True}, \text{False}\}$.

Initialize variables as follows: $Y = 0$, $L = 0$, $S = 0$, $\mu = 1.25/\|M'\|_2$, $\rho = 1.1$, and $converged$ = False. Then, we begin the following iteration:
`While` (not $converged$):
1. Update the values of L and S:
   $L = \mathcal{D}_{\mu^{-1}}(M' - S + \mu^{-1}Y)$
   $S =$ `Find_Optimal_S`$(M', L, \tilde{\epsilon}, Y, \mu)$
   (Described in the text.)
2. Update the Lagrange multipliers:
   $Y = Y + \mu(M' - L - S)$
3. Check for convergence:
   $\Delta = \|M' - L - S\|_F / \|M'\|_F$
   `If`$(\Delta < tol)$:
      $converged$ = True

`Return` $\hat{L} = L$, $\hat{S} = \mathcal{S}_{\tilde{\epsilon}}(S)$, $\hat{Z} = M - \hat{L} - \hat{S}$

Algorithm 1: **eRPCA**

coupling between these groupings. Thus, each grouping of three terms can be minimized independently (using straightforward algebra) and then summed together to obtain the global minimum for each entry in S.

E. Correlated Time Series Analysis

Consider a *sensor network* in which each of the N_s nodes is a sensor measuring a real vector-valued time series. In the sense of Information Assurance on a computer network [21], [22], the measured time series might represent port activity, CPU load, packet rates, password failures, *etc*. Thus, the i-th sensor collects a vector-valued measurement of dimension l_i and duration n. We then construct a signal matrix $Y \in \mathbb{R}^{m\times n}$ by concatenating all the vector-valued time series from all the nodes where the number of rows in Y is $m = \sum_{i=1}^{N_s} l_i$, and the number of columns in Y is the number of discrete time intervals for which data were collected. The matrix Y therefore has rows that are time traces of a particular quantity of interest (the CPU load of node i, for example), and has columns that are spatial snapshots of the state of the network at a particular time.

For ease of exposition, we will consider normalized data matrices where

$$Y = (n-1)^{-\frac{1}{2}} \text{diag}[\sigma_1^{-1}, \cdots, \sigma_m^{-1}] \left(\tilde{Y} - \mu_{\tilde{Y}} \mathbf{1}^T\right)$$

for some original raw data matrix $\tilde{Y}$ with row-wise mean $\mu_{\tilde{Y}} \in \mathbb{R}^{m\times 1}$, and row-wise standard deviation $\sigma_{\tilde{Y}} \in \mathbb{R}^{m\times 1}$, and $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is a vector of all ones. For the normalized data matrix, Y, the sample Pearson correlation matrix can be written as YY^T [23].

F. Latent Signal Models

We argue that if the network data Y is patterned (meaning that a sparse subset of nodes exhibits an anomalous correlation), then YY^T may be modeled as Wishart $\mathcal{W}_m(M_0, n)$, with M_0 a structured matrix of the form

$$M_0 = L_0 + S_0 + \Sigma_0, \tag{15}$$

where the matrix L_0 is low-rank, the matrix S_0 is sparse, and the matrix Σ_0 is diagonal. Performance guarantees for PCP require that YY^T be the sum of low-rank and sparse matrices, but no Wishart matrix has this property. Thus, we require modifications to PCP to allow for point-wise inequality constraints to accommodate the Wishart fluctuations of YY^T around the matrix M_0.

The structure of the correlation matrix M_0 given in (15) is based on our ansatz that the first-order matrix Y obeys the following latent time series model:

$$Y = AU + BV + N, \tag{16}$$

under the assumption that U, V, and N are independent Gaussian matrices loaded with normal $\mathcal{N}(0,1)$ random variables. Then, YY^T is Wishart $\mathcal{W}_m(M_0, n)$ with

$$M_0 = A\Sigma_{UU}A^T + B\Sigma_{VV}B^T + \Sigma_{NN}\,, \tag{17}$$

for diagonal matrices Σ_{UU}, Σ_{VV}, and Σ_{NN}. If A is low-rank and B is sparse, then M is the sum of a low-rank matrix $L_0 := A\Sigma_{UU}A^T$, sparse matrix $S_0 := B\Sigma_{VV}B^T$, and a diagonal matrix $\Sigma_0 := \Sigma_{NN}$. This is the model we would like to fit to YY^T, with point-wise error constraints allowing for Wishart fluctuations.

The decomposition model in (16) states that Y is a linear combination of mutually uncorrelated time traces that represent the core contributing sources to each of the measured time series. Our approach is to simultaneously determine those nodes whose behavior is well-explained by the behavior of all their peers, as well as those nodes that appear to be simultaneously affected by an unusual underlying process that is outside the mainstream.

Classically, *Principal Component Analysis* (PCA) provides the best low-rank approximation (in the sense of the Frobenius norm) to a given matrix [24]. Unfortunately, it is well-known that PCA suffers when outliers are present — a single outlier can skew the approximated low-rank subspace arbitrarily far away from the true low-rank subspace [25]. Principal Component Pursuit allows for careful teasing apart of sparse outliers so that the remaining low-rank approximation is faithful to the true low-rank subspace describing the raw data [4], [5], [15].

At first blush, one may attempt to apply PCP directly to the first order data matrix Y. There are, however, several advantages to analyzing the second order correlation matrix, YY^T, instead. First, for many problems of interest, $m \ll n$ so

that the matrix YY^T is much smaller in size than the matrix Y. This is advantageous in cyber domains [21], [22] where it is infeasible to communicate the entire data matrix Y across the network. Second, studying YY^T provides some measure of noise mitigation as compared to studying Y. For example, if N consists of uncorrelated and identically distributed draws from a zero mean, unit variance Gaussian distribution, then $\frac{1}{n}NN^T$ is Wishart $W_m(I,n)$ with diagonal entries of unit mean and variance $\frac{1}{n}$, and off-diagonal entries of zero mean and variance $\frac{1}{n}$. In effect, the matrix Σ_{NN} is an identity matrix with Wishart fluctuations that are smaller than the fluctuations in the original data Y.

Our analysis decomposes $M = YY^T$ into a *low-rank* part that indicates the presence of a pervasive low-dimensional pattern affecting the entire network, and a *sparse* part that indicates sparse correlations between a few nodes that are anomalous when compared to the ambient background correlation.

In this approach, the role of the projection operator, $\mathcal{P}_\Omega$, bears further comment. Recall from our earlier discussion on the latent signal model that the error matrix Z_0 is Wishart $\mathcal{W}_m(\Sigma_{NN}, n)$ for diagonal matrix Σ_{NN}. Any point-wise control of the entries in Z_0 should allow for larger point-wise errors on the diagonals where Σ_{NN} is large. Since Σ_{NN} is unknown, we proceed by removing the diagonal entries from consideration by adding the diagonal to the set of unobserved entries in Ω. That is, we expect M to be close to the sum of a low-rank matrix and a sparse matrix *on the off-diagonal entries* and allow the matrix completion algorithm to provide the unknown entries on the diagonal.

In the context of sensor networks, the introduction of entry-wise error control in (6) is motivated by the reality that we may receive data from heterogeneous sensors, and consequently, we may wish to ascribe different error tolerances to each individual measurement, to each sensor, or to each sensor pair (for the second-order matrix).

We emphasize that the proposed algorithm is intended to reveal anomalous temporal correlations between time signals; a situation where the signals at a small number of nodes all contain a component that is not felt anywhere else on the network. Large values in $\hat{S}$ returned by the algorithm indicate that during the time interval under examination, a sparse correlation occurred on that set of sensors. Moreover, the algorithm cannot identify at which moment during that time interval the anomaly occurred. Indeed, the anomaly detected by the algorithm may arise from a diffuse signal that occurs unobtrusively throughout the entire time interval and is therefore not localized in time.

III. Results and Discussion

A. Tests on Synthetic Data

To judge the performance of our algorithm, we compare it to the Frobenius norm based formulation (5) presented in [5]. We closely follow the test procedure in [5] by constructing a noise matrix Z_0, a low-rank matrix L_0, and a sparse matrix S_0 as follows. The noise matrix $Z_0 \in \mathbb{R}^{n\times n}$ has entries which are i.i.d. $N(0,\sigma^2)$ for a prescribed noise standard deviation σ. We construct the rank-r matrix $L_0 = U_0V_0$ using $U_0 \in \mathbb{R}^{n\times r}$ and $V_0 \in \mathbb{R}^{r\times n}$ with i.i.d. entries from $N(0,\sigma_n^2)$ where, as in [5], we choose $\sigma_n = 10\frac{\sigma}{\sqrt{n}}$. As noted in [5], this choice for σ_n ensures that the singular values of L_0 are large compared to the singular values of Z_0. Finally, we construct the sparse anomaly matrix $S_0 \in \mathbb{R}^{n\times n}$ with independently distributed entries, each being zero with probability $1-\rho_s$, and uniform i.i.d. in the range $[-5,5]$ with probability ρ_s. In all our comparisons, we average over twenty Monte Carlo realizations of the problem data.

To compare (5) and (6), we use an accelerated proximal gradient method [26] to solve a dual version of (5) as

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1 + \frac{1}{2\mu_d}\|M-L-S\|_F^2 \qquad (18)$$

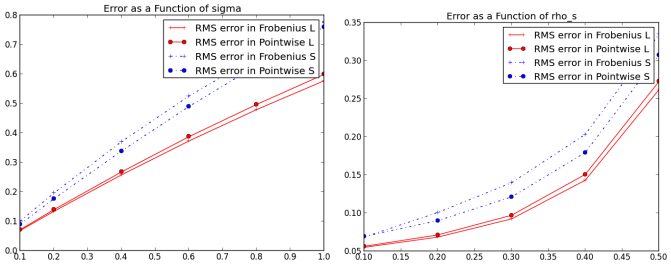
(see [5] for details), and Algorithm 1 for solving (6).

As we use a fast ADMM method for (6), the computational cost of our method is quite competitive when compared to the proximal gradient method for (5). Quantitative comparisons of computational time are somewhat difficult, as the proximal gradient solver we use is written in Matlab [26], and our ADMM solver is written in C++. On a virtual machine using a 2 GHz QEMU Virtual CPU version 1.0 processor, the run time for the two algorithms is generally within a factor of two of each other, with the more efficient algorithm changing based upon the exact problem one is solving. For one example, the ADMM algorithm uses approximately 42 seconds of CPU time, while the proximal gradient algorithm uses approximately 69 seconds. On another example, the ADMM algorithm uses approximately 43 seconds of CPU time, while the proximal gradient algorithm uses approximately 25 seconds of CPU time. Of course, these numbers vary based upon the iteration count for the two algorithms. The most expensive computation for both algorithms is an SVD decomposition at each iteration, so it is not surprising that their computational costs are similar. We also observe that both algorithms can be accelerated using partial SVD algorithms.

Both (5) and (6) require judicious choice of parameters to recover L_0 and S_0 well. In particular, the dual version of (5) requires a choice of coupling constant, μ_d, between the objective and the Frobenius constraint, and we follow [5] by choosing $\mu_d = \sqrt{2n}\sigma$. Similarly, eRPCA requires choice of the point-wise constraint matrix $\tilde{\epsilon}$. The goal of eRPCA is to allow the user flexibility in setting $\tilde{\epsilon}$ based upon the structure of the problem. In particular, the user may set every entry of $\tilde{\epsilon}$ differently to encode the knowledge they have for their problem. As (5) does not afford this flexibility, we choose a fixed value for all of the entries in $\tilde{\epsilon}$ as $\tilde{\epsilon}_{ij} = 0.67\sigma$. In this way, half of the total probability mass for the Gaussian noise is in the range $[-\tilde{\epsilon}_{ij}, \tilde{\epsilon}_{ij}]$ and half of the probability mass is outside that range.

Figure 1 shows a comparison between (5) and (6). As in [5], we choose $n = 200$, $\sigma = 0.1$, and $\rho_s = 0.2$ as our base case. To judge the performance of the algorithms, we use the RMS error of the recovered $\hat{L}$ and $\hat{S}$ as $\|L_0 - \hat{L}\|_F/n$ and $\|S_0 - \hat{S}\|_F/n$ respectively. Figure 1(a) shows the recovery error as σ varies and Figure 1(b) shows the recovery error as ρ_s varies. As can be observed, the point-wise algorithm

performs quite closely to the Frobenius algorithm over the range of parameters tested. Perhaps the point-wise algorithm is slightly better in recovery of S_0, and slightly worse in the recovery of L_0 but neither algorithm is clearly superior.

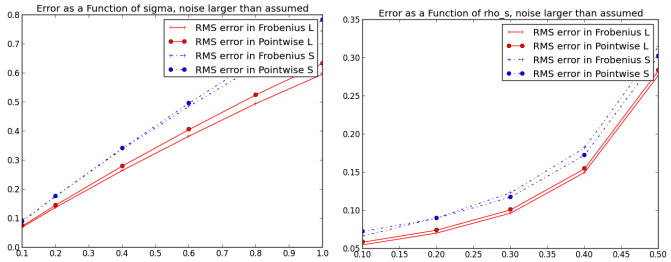


(a) σ (b) ρ_s

Fig. 1. These figures show the RMS error in the recovery of L_0 and S_0, as a function of σ on the left and ρ_s on the right, when the correct σ is reported to both the Frobenius algorithm and the eRPCA algorithm.

It is interesting to note that, as defined above, μ_d and $\tilde{\epsilon}$ both depend on knowledge of the noise level σ, which is perhaps difficult to judge in measured data. Accordingly, we have generalized the testing procedure in [5] by considering the sensitivity of the algorithms to inaccurately known noise.

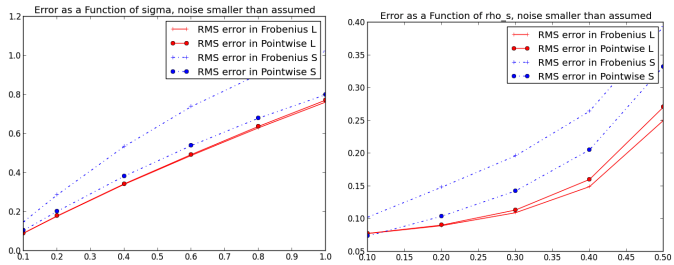
Figure 2 shows a similar experiment to Figure 1 except the true σ is *twice as large* as the σ reported to the algorithm for setting μ or $\tilde{\epsilon}$. In this case, the two algorithms are also roughly equivalent; the noise is sufficiently large that neither constraint can properly account for the large noise entries. Accordingly, the noise corrupts both $\hat{L}$ and $\hat{S}$ for both algorithms.



(a) σ (b) ρ_s

Fig. 2. These figures show the RMS error in the recovery of L_0 and S_0, as a function of σ on the left and ρ_s on the right, when an incorrect σ is reported to both the Frobenius algorithm and the eRPCA algorithm. In this case, the actual noise is two times *larger* than what would be indicated by the reported σ.

Figure 3, on the other hand, tells a different story. Figure 3 is a similar experiment to Figure 1 except, in this case, the true σ is half the size of the σ reported to the algorithms for setting μ or $\tilde{\epsilon}$. In this case, the recovery of L_0 is roughly equivalent between the two algorithms, but the recovery of S_0 is *appreciably better* for the point-wise constraint. Since the Frobenius constraint is a global error measure, the freedom it provides can be used to both ameliorate noise and to mask entries in S_0 *globally*. The point-wise error budget, on the other hand, is specified for each entry of the matrix individually. Accordingly, any entry $S_{0_{ij}}$ that is larger than $\tilde{\epsilon}_{ij}$ cannot be masked by the error constraint, no matter how small the noise happens to be in other entries.



(a) σ (b) ρ_s

Fig. 3. These figures show the RMS error in the recovery of L_0 and S_0, as a function of σ on the left and ρ_s on the right, when an incorrect σ is reported to both the Frobenius algorithm and the eRPCA algorithm. In this case, the actual noise is half the size of what would be indicated by the reported σ.

We now observe that the point-wise constraints in (6) can be used to encode additional information that goes beyond having a constant $\tilde{\epsilon}$. In Figure 4, we show an example where the noise σ actually varies from entry to entry in the matrix. For this example, we think of each entry as a sensor measuring some quantity of interest. The sensors have different known noise characteristics, and we wish to avail ourselves of this information. In particular, 90% of the entries have the indicated σ value, while the remaining 10% have a σ which is sixteen times larger. Unlike the point-wise algorithm, the Frobenius-constrained algorithm is not designed to utilize this auxiliary knowledge. Therefore, it is not surprising that the Frobenius algorithm has difficulty recovering S_0 since it has no way to discriminate between high noise and low noise sensors.

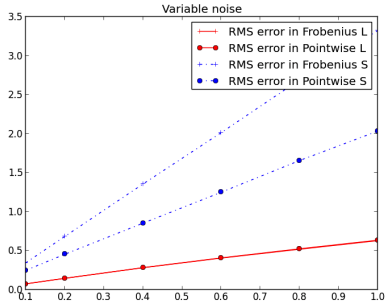


Fig. 4. This figure shows the RMS error in the recovery of L and S in a case where the noise changes for different entries. Ninety percent of the entries have the indicated σ value, while the remaining ten percent have a σ which is sixteen times larger. Unlike the eRPCA algorithm, the Frobenius algorithm does not utilize this knowledge.

B. Suggestions for choosing $\tilde{\epsilon}$

The previous section showed a comparison between the point-wise and Frobenius constraints on problems with Gaussian noise using Frobenius norm metrics. One of the key parameters in the eRPCA algorithm is $\tilde{\epsilon}$, and choosing $\tilde{\epsilon} = 0.67\sigma$ provides a competitive algorithm. On the other hand, our common usage of the point-wise algorithm has somewhat

different goals, so here we will provide some suggestions as to how the algorithm might be used.

Classically, the goal of matrix recovery algorithms is to recover L_0 by removing the corruptions introduced by S_0. We take the opposite approach. As discussed in Section II-F, we consider the low rank L_0 to be generated by a collection of ubiquitous background processes that mask the sparsely correlated anomalies in S_0 that we seek to uncover. This focus can be be seen in the previous section, where our recovery of S_0 is almost always better using the point-wise error constraint instead of the Frobenius constraint.

Furthermore, we are often less concerned about the values in the recovered $\hat{S}$ than we are about detecting the *support* of S_0. As we will demonstrate in the next section, the support of $\hat{S}$ in our second-order matrices is indicative of the sparse correlations that we wish to detect. With this in mind, we often choose $\tilde{\epsilon}$ to be quite large. The idea is to capture the majority of the noise in the error constraint, thereby making the support of $\hat{S}$ less contaminated by noise. Of course, there is a trade-off. An $\tilde{\epsilon}$ which is very large for many entries of M_0 not only ameliorates the noise, it also allows $\hat{L}$ to stray far from L_0. Accordingly, our heuristic is to choose $\tilde{\epsilon}$ larger than our estimate for the noise, but not so large that it tends to dominate L_0.

C. Tests on Measured Data

In this section, we apply our algorithms for detecting anomalies to time traces recorded from the Abilene Internet2 network backbone [27]. The geographical location of the nodes that participate in the Abilene network can be found in the map shown in Figure 5(c). Specifically, we use Abilene data collected over a period of 350 days that records packet throughput over one hour intervals for 28 links. Accordingly, our full data set is $24 \cdot 350 = 8400$ measurements over each of the 28 links. We divide that data into 25 two week windows and examine each window individually, giving rise to a collection of matrices $\{Y_i \in \mathbb{R}^{28\times 336}, i \in \mathbb{N}, 1 \leq i \leq 25\}$.

To demonstrate our algorithms, we attempt to detect native patterns in the Abilene data using only a second-order analysis of each of the $Y_i Y_i^T$. Since the Abilene data is not labeled with anomalies, we then allow ourselves to cross-validate our second-order analysis by an *a posteriori* examination of the first-order data. Interestingly, this test is close to our view of the intended fielded usage of our work. We view our approach as a pre-processing step that automatically detects anomalous sensors in voluminous real world data, and then presents to the user a condensed view of the network performance and anomalies that highlights areas that require further analysis. False alarms are mitigated since each anomaly is only detected and reported when it occurs at multiple sensors.

Distributed patterns can be detected in several places in the raw Abilene data. The first example of such a pattern is shown in Figure 5. In this case, our methods detected that three Abilene links experienced an anomaly over the course of the two week window without any predefined pattern template. Further examination of the the first order data reveals an anomaly that appeared over the course of a three day period. We are able to isolate the sensors experiencing this pattern using only second order information.

In the second example shown in Figure 6, the data set happens to contain two separate anomalies. The inference that these anomalies are two different events is evinced by the fact that each pair of nodes are sparsely correlated with each other, but there is no sparse correlation between the two pairs.

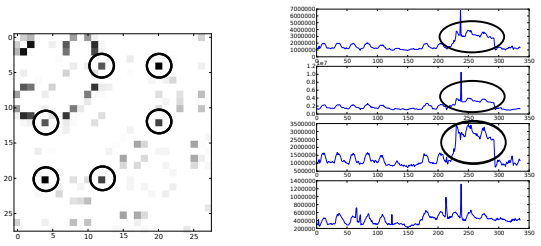
These anomalies were detected using a very special projection operator $\mathcal{P}_\Omega$. Recall that the entries of M are the correlations of each pairing of time series produced at all the nodes. In order to fully observe M, one must compute a correlation for *every pair* of rows in Y even if the nodes are widely separated in the network. On the other hand, *a partially observed M can be produced using only time series that exist at neighboring nodes in G*. The time series in the Abilene data happen to correspond to packet rates across unidirectional Internet links. Therefore, each node possesses the time series for every link that either originates or terminates at that node, and consequently, *a partially observed M can be computed without the transmission of any time series.* This is precisely the $\mathcal{P}_\Omega$ used for the analysis in Figure 6. Each node uses only the transmitted or received time series it already has on hand based upon its normal operating procedure. Note, the results of the distributed correlations still need to be collected at some central node for processing, but as $m \ll n$, and a correlation coefficient is a single scalar, the size of this dataset is small when compared to the original time series.

This idea allows us to recover L_0 and S_0 using only those entries in M that can be computed locally. In other words, *every correlation we used was available at some sensor on the network without requiring the communication of any additional time series information*. The algorithm only required a highly compressed form of the data natively available on the network, and was therefore extremely efficient in its use of network resources.

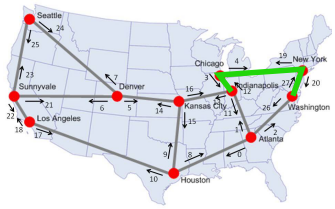
Finally, in Figure 7 we show a comparison of the anomaly detection capabilities of the Frobenius norm algorithm (5) and the point-wise error algorithm (6) on measured data. As is suggested by our tests on synthetic data, for a tuned noise parameter σ, the performance of the two algorithms is quite similar. As this is measured data there is no particular theoretical principle for selecting σ so we empirically set $\sigma = 0.1$ and choose the parameters of both algorithms as in Section III-A. Figure 7 (a) and (c) demonstrate that the same anomaly is detected by both algorithms for this choice of σ. On the other hand, when one uses a suboptimal σ value (in this case $\sigma = 0.4$) then one can see in Figure 7 (b) and (d) that the point-wise algorithm still detects the original anomaly, while the Frobenius algorithm doesn't detect one of the nodes participating in the anomaly.

IV. Conclusions and Future Directions

We have presented a mathematical framework for detecting anomalous correlations on networks. We have derived an algorithm for decomposing raw data into its pattern primitives — a low-rank description of network-wide influences, as well as a sparse description of anomalous correlations influencing a few nodes.

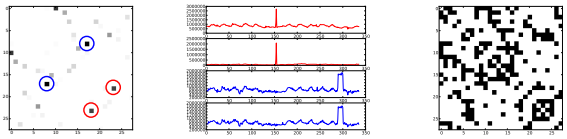


(a) The sparse correlation matrix $\hat{S}$. (b) The signals in the raw data.



(c) The map of the connectivity of the Abilene network with the detected pattern highlighted.

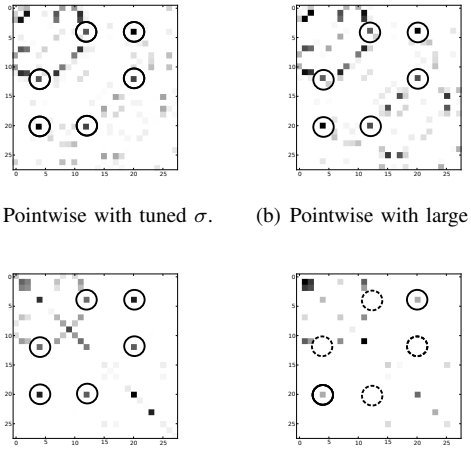
Fig. 5. Here we show an example of a pattern detected in raw Abilene data. (a) A sparse $\hat{S}$ matrix computed by way of the eRPCA algorithm with strong off-diagonal entries. (b) The three corresponding time series on top with a fourth uncorrelated time series on the bottom for reference. The pattern that has been detected is the three day traffic anomaly highlighted by the black ovals. Neither the form of the pattern (a three day traffic anomaly) nor the actual time series were needed for the analysis. As indicated in (c), the anomalous links form a continuous directed path from Indianapolis to Washington, D.C.



(a) The sparse correlation matrix $\hat{S}$. (b) The signals in the raw data. (c) The support of the $\mathcal{P}_\Omega$ used in this calculation.

Fig. 6. Here we show another example of a pattern detected in raw Abilene data. (a) A sparse $\hat{S}$ matrix computed by way of the eRPCA algorithm again with strong off-diagonal entries. Two pairs of the sparse anomalies are labeled with red and blue circles. (b) The two pairs of times series corresponding to the labeled anomalies in (a), with the same color scheme. (c) The $\mathcal{P}_\Omega$ used for this calculation. Every inner product needed to form YY^T was available at some sensor on the network without requiring the communication of any additional time series information

As a final remark, we note that our current work applies matrix decomposition methods to the correlation matrices of the form YY^T, but there is nothing preventing the application of these ideas to a much wider problem domain. In particular, further efforts will extend the linear relationships implied by the correlation matrix to fully nonlinear similarity models that describe arbitrary functional dependence. Examples of other possible similarity measures include *mutual information* [28], *kernel maps* [29], *copulas* [30], and *maximal information coefficients* [31].



(a) Pointwise with tuned σ. (b) Pointwise with large σ. (c) Frobenius with tuned σ. (d) Frobenius with large σ.

Fig. 7. Here we show a comparison of the anomaly detection capabilities of the Frobenius norm algorithm (5) and the point-wise error algorithm (6) on measured data. (a) uses the point-wise error bound and a tuned noise σ (for completeness this figure is repeated from Figure 5). (b) uses the point-wise error bound and a σ which is four times larger. (c) uses a Frobenius bound and a tuned noise σ. (d) uses a Frobenius bound and a σ which is four times larger. As suggested by the tests on synthetic data, the anomaly is detected in all cases except for (d) where the Frobenius bound has problems with misspecified noise, as shown by the dashed circles. On the other hand, the point-wise results in (a) and (b) are largely unaffected by the misspecified noise.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions for improving the paper. All authors gratefully acknowledge funding for this research from the Air Force Office of Scientific Research under STTR contract FA9550-10-C-0090, and from the IRAD program at Numerica Corporation.

APPENDIX
PROOF OF THEOREM 2.1

The proof of Theorem 2.1 follows very closely the proof of Theorem 2 in [5] for the case of a Frobenius norm constraint. Even so, the main architecture of the proof of Theorem 2.1 is provided here for completeness. The proof considers the case of fully observed matrices, and then a simple argument extends the result to partially observed matrices with tighter requirements on sparsity. Throughout the proof we use the notation $\mathcal{P}_A$ to denote the projection operator onto the subspace A.

We begin by stating the definition for *incoherence* as given in [18]. Recovery will be dependent on the matrix L_0 being sufficiently incoherent.

Definition A.1 (Incoherence conditions): Write the singular value decomposition of $L_0 \in \mathbb{R}^{n\times n}$ as

$$L_0 = U\Sigma V^* = \sum_{i=1}^{r} \sigma_i u_i v_i^*,$$

where r is the rank of the matrix, $\sigma_1, \cdots, \sigma_r$ are the singular values, and $U = [u_1, \cdots, u_r]$, $V = [v_1, \cdots, v_r]$ are the matrices of left- and right-singular vectors. Then, the incoherence parameter μ states that

$$\max_i \|U^* e_i\|^2 \leq \frac{\mu r}{n}, \qquad \max_i \|V^* e_i\|^2 \leq \frac{\mu r}{n}, \tag{19a}$$

and

$$\|UV^*\|_\infty \leq \frac{\sqrt{\mu r}}{n}\,. \tag{19b}$$

Corollary 2.7 in [18] uses this incoherence condition along with Theorem 4.1 in [2] to establish that with high probability, $\|\mathcal{P}_\Psi \mathcal{P}_T\| \leq 1/2$ where Ψ is the subspace of matrices supported on support(S_0), and T is the subspace of matrices with the same row space or column space as L_0.

Next, [18] shows that if a certificate W with specific properties can be provided, then the true solution (L_0, S_0) is the optimal solution of the convex program

$$\begin{aligned} &\min_{L,S} \|L\|_* + \lambda \|S\|_1 \\ &\text{subject to} \quad L + S = M. \end{aligned} \tag{20}$$

Lemma A.2 (Lemma 3 in [5], Lemma 2.5 in [18]): Assume $\|\mathcal{P}_\Psi \mathcal{P}_T\| \leq 1/2$, and take $\lambda < 1$. Suppose that there exists W such that

$$\begin{cases} W \in T^\perp \\ \|W\| < 1/2 \\ \mathcal{P}_\Psi(UV^* - \lambda \text{sgn}(S_0) + W)\|_F \leq \lambda/4 \\ \|\mathcal{P}_{\Psi^\perp}(UV^* + W)\|_\infty < \lambda/2\,. \end{cases} \tag{21}$$

Then, the pair (L_0, S_0) is the unique optimal solution to (20).

Proof: Proof is provided in section 2.3 in [18]. ∎

In the next important step, Lemma 2.8 and Lemma 2.9 in [18] show that, with high probability, the necessary certificate $W = W^L + W^S$ can be constructed using the "golfing scheme" described in [32]. Importantly, the construction of this certificate does not depend on the form of the constraint in (20); it depends only on the properties of L_0 and S_0.

The main result from [18] that we will need for the proof of Theorem 2.1 is the guarantee that with high probability a dual certificate W satisfying (21) can be constructed.

We will also need two more Lemmas proved in [5]. First, to set notation, we say that for any matrix pair $X = (L, S)$:

$$\begin{aligned} \|X\|_\diamond &:= \|L\|_* + \lambda \|S\|_1, \\ \|X\|_F^2 &:= \|L\|_F^2 + \|S\|_F^2, \\ \mathcal{P}_\Gamma(X) &:= \left(\frac{L+S}{2}, \frac{L+S}{2} \right), \end{aligned}$$

and

$$(\mathcal{P}_T \times \mathcal{P}_\Psi) X := (\mathcal{P}_T L, \mathcal{P}_\Psi S).$$

Lemma A.3 (Lemma 5 in [18]): Assume that $\|\mathcal{P}_\Psi \mathcal{P}_T\| \leq 1/2$ and $\lambda \leq 1/2$. Suppose that there exists a dual certificate W satisfying (21) and write $\Lambda = UV^* + W$. Then for any perturbation, $H = (H_L, H_S)$ obeying $H_L + H_S = 0$,

$$\begin{aligned} \|X_0 + H\|_\diamond \geq &\|X_0\|_\diamond + (3/4 - \|\mathcal{P}_{T^\perp}(\Lambda)\|)\, \|\mathcal{P}_{T^\perp}(H_L)\|_* \\ &+ (3\lambda/4 - \|\mathcal{P}_{\Psi^\perp}(\Lambda)\|_\infty)\, \|\mathcal{P}_{\Psi^\perp}(H_S)\|_1\,. \end{aligned}$$

Lemma A.4 (Lemma 6 in [18]): Suppose that $\|\mathcal{P}_T \mathcal{P}_\Psi\| \leq 1/2$. Then for any pair $X := (L, S)$,

$$\|(\mathcal{P}_T \times \mathcal{P}_\Psi)(X)\|_F^2 \leq 4 \|\mathcal{P}_\Gamma (\mathcal{P}_T \times \mathcal{P}_\Psi)(X)\|_F^2\,.$$

We are now prepared to state our main proposition:

Proposition A.5 (Modification of Proposition 4 in [5]): Assume $\|\mathcal{P}_\Psi \mathcal{P}_T\| \leq 1/2$, $\lambda \leq 1/2$, and that there exists a dual certificate W satisfying (21). Let $\hat{X} = (\hat{L}, \hat{S})$ be the solution to (6), and $X_0 := (L_0, S_0)$; then $\hat{X}$ satisfies

$$\|X_0 - \hat{X}\|_F \leq (8\sqrt{5}n + \sqrt{2})\delta\,.$$

Proposition A.5 implies Theorem 2.1, since under the conditions of Theorem 2.1, the results cited in [18] show that with high probability there indeed exists a dual certificate W that satisfies (21), and Corollary 2.7 of [18] proves $\|\mathcal{P}_\Psi \mathcal{P}_T\| \leq 1/2$ as well. Importantly, we see that we can very quickly use this result to demonstrate stable recovery for the case of a *partially* observed M_0. We do so by regarding the unobserved entries of M_0 as corruptions (non-zero entries in S_0), and then applying the theorem for fully observed matrices where the requirements on the sparsity of the support of S_0 must also include the support of the unobserved entries.

Finally, we proceed to the proof of Proposition A.5 which uses the same arguments as [5].

Proof: Our algorithm is stable if the error difference between truth, $X_0 := (L_0, S_0)$, and our recovery, $\hat{X} := (\hat{L}, \hat{S})$, is bounded by the size of the noise. To quantify this error, we define

$$H := (H_L, H_S) := \hat{X} - X_0\,.$$

Since $\hat{X}$ minimizes the optimization problem in (6), and X_0 is also a feasible solution to the optimization, it is necessarily true that

$$\|\hat{X}\|_\diamond \leq \|X_0\|_\diamond\,. \tag{22}$$

Second, using the triangle inequality,

$$\begin{aligned} \|\hat{L} + \hat{S} - L_0 - S_0\|_F &\leq \|\hat{L} + \hat{S} - M\|_F + \|L_0 + S_0 - M\|_F \\ &\leq \|\hat{L} + \hat{S} - M\|_1 + \|L_0 + S_0 - M\|_1 \\ &\leq \|\epsilon\|_1 + \|Z_0\|_1 \\ &\leq 2\delta\,. \end{aligned} \tag{23}$$

Our end goal is to bound the size of

$$\|H\|_F^2 = \|H_L\|_F^2 + \|H_S\|_F^2 = \|\hat{L} - L_0\|_F^2 + \|\hat{S} - S_0\|_F^2$$

by a term that goes to zero as the size of the noise goes to zero. To achieve this, we seek to control

$$\|H\|_F^2 = \|H^\Gamma\|_F^2 + \|H^{\Gamma^\perp}\|_F^2\,.$$

We begin by bounding the first term using (23) as follows:

$$\begin{aligned} \|H^\Gamma\|_F^2 &= \left\| \frac{H_L + H_S}{2} \right\|_F^2 + \left\| \frac{H_L + H_S}{2} \right\|_F^2 \\ &= \frac{1}{2} \|H_L + H_S\|_F^2 \\ &= \frac{1}{2} \|\hat{L} - L_0 + \hat{S} - S_0\|_F^2 \\ &\leq 2\delta^2\,. \end{aligned} \tag{24}$$

All that remains is to bound the term $\|H^{\Gamma^\perp}\|_F^2$. Again, we divide this term into two terms using orthogonal projections:

$$\begin{aligned}\|H^{\Gamma^\perp}\|_F^2 &= \|\mathcal{P}_T H_L^{\Gamma^\perp}\|_F^2 + \|\mathcal{P}_{T^\perp} H_L^{\Gamma^\perp}\|_F^2 \\ &\quad + \|\mathcal{P}_\Psi H_S^{\Gamma^\perp}\|_F^2 + \|\mathcal{P}_{\Psi^\perp} H_S^{\Gamma^\perp}\|_F^2 \\ &= \|(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp})\|_F^2 \\ &\quad + \|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F^2 , \end{aligned} \tag{25}$$

and bound the terms independently.

First, consider $\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F^2$. Let W be a dual certificate satisfying (21). Then, $\Lambda := UV^* + W$ obeys $\|\mathcal{P}_{T^\perp}(\Lambda)\| \leq \frac{1}{2}$ and $\|\mathcal{P}_{\Psi^\perp}(\Lambda)\|_\infty \leq \frac{\lambda}{2}$. By the triangle inequality, and the bound $\|\hat{X}\|_\diamond \leq \|X_0\|_\diamond$ obtained in (22), we have

$$\begin{aligned}\|X_0 + H^{\Gamma^\perp}\|_\diamond &\leq \|X_0 + H\|_\diamond + \|H^\Gamma\|_\diamond \\ &= \|\hat{X}\|_\diamond + \|H^\Gamma\|_\diamond \\ &\leq \|X_0\|_\diamond + \|H^\Gamma\|_\diamond . \end{aligned} \tag{26}$$

Also, by Lemma A.3, we have

$$\begin{aligned}&\|X_0 + H^{\Gamma^\perp}\|_\diamond \\ \geq &\|X_0\|_\diamond + (3/4 - \|\mathcal{P}_{T^\perp}(\Lambda)\|)\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* \\ &+ (3\lambda/4 - \|\mathcal{P}_{\Psi^\perp}(\Lambda)\|_\infty)\|\mathcal{P}_{\Psi^\perp}(H_S^{\Gamma^\perp})\|_1 \\ \geq &\|X_0\|_\diamond + (3/4 - 1/2)\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* \\ &+ (3\lambda/4 - \lambda/2)\|\mathcal{P}_{\Psi^\perp}(H_S^{\Gamma^\perp})\|_1 \\ \geq &\|X_0\|_\diamond + 1/4\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* + \lambda/4\|\mathcal{P}_{\Psi^\perp}(H_S^{\Gamma^\perp})\|_1 \end{aligned} \tag{27}$$

where we have used the assumptions $\|\mathcal{P}_{T^\perp}(\Lambda)\| \leq \frac{1}{2}$ and $\|\mathcal{P}_{\Psi^\perp}(\Lambda)\|_\infty \leq \frac{\lambda}{2}$. Rearranging (27) yields

$$\begin{aligned}1/4\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* + \lambda/4\|\mathcal{P}_{\Psi^\perp}(H_S^{\Gamma^\perp})\|_1 \\ \leq \|X_0 + H^{\Gamma^\perp}\|_\diamond - \|X_0\|_\diamond . \end{aligned} \tag{28}$$

Taking the inequalities in (26) and (28) together implies that

$$\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* + \lambda\|\mathcal{P}_{\Psi^\perp}(H_S^{\Gamma^\perp})\|_1 \leq 4\|H^\Gamma\|_\diamond . \tag{29}$$

Note that by definition of H^Γ,

$$\begin{aligned}\|H^\Gamma\|_F &= \left(\left\| \frac{H_L + H_S}{2} \right\|_F^2 + \left\| \frac{H_L + H_S}{2} \right\|_F^2 \right)^{\frac{1}{2}} \\ &= \sqrt{2} \left\| \frac{H_L + H_S}{2} \right\|_F = \sqrt{2} \left\| H_L^\Gamma \right\|_F \\ &= \frac{1}{\sqrt{2}} \left(\left\| H_L^\Gamma \right\|_F + \left\| H_S^\Gamma \right\|_F \right) , \qquad \text{(using } H_L^\Gamma = H_S^\Gamma\text{)} \end{aligned}$$

so that we can write,

$$\left\| H_L^\Gamma \right\|_F + \left\| H_S^\Gamma \right\|_F = \sqrt{2}\|H^\Gamma\|_F . \tag{30}$$

Therefore,

$$\begin{aligned}&\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F \\ &\leq \|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_F + \|\mathcal{P}_{\Psi^\perp}(H_S^{\Gamma^\perp})\|_F \\ &\leq \|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* + \lambda\sqrt{n}\|\mathcal{P}_{\Psi^\perp}(H_S^{\Gamma^\perp})\|_1 \\ &\leq 4\sqrt{n}\|H^\Gamma\|_\diamond \qquad \text{(using (29))} \\ &= 4\sqrt{n}(\|H_L^\Gamma\|_* + \lambda\|H_S^\Gamma\|_1) \\ &\leq 4n(\|H_L^\Gamma\|_F + \|H_S^\Gamma\|_F) \\ &= 4\sqrt{2}n\|H^\Gamma\|_F \leq 8n\delta , \qquad \text{(using (30) and (24))} \end{aligned}$$

from which we may write

$$\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F^2 \leq 64n^2\delta^2 . \tag{31}$$

Now, we proceed to derive a bound for the first term in (25). By Lemma A.4,

$$\|(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp})\|_F^2 \leq 4\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp})\|_F^2 . \tag{32}$$

Also, since

$$\mathcal{P}_\Gamma(H^{\Gamma^\perp}) = 0 = \mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp}) + \mathcal{P}_\Gamma(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})$$

implies

$$\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp}) = -\mathcal{P}_\Gamma(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp}),$$

we have

$$\begin{aligned}\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp})\|_F &= \|\mathcal{P}_\Gamma(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F \\ &\leq \|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F . \end{aligned} \tag{33}$$

Combining the previous two inequalities in (32) and (33), we have

$$\|(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp})\|_F^2 \leq 4\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F^2 ,$$

which together with (31) yields

$$\|(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp})\|_F^2 \leq 4 \cdot 64n^2\delta^2 . \tag{34}$$

Combining the bounds obtained in (31) and (34), we have

$$\begin{aligned}\|H^{\Gamma^\perp}\|_F^2 &= \|(\mathcal{P}_T \times \mathcal{P}_\Psi)(H^{\Gamma^\perp})\|_F^2 + \|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Psi^\perp})(H^{\Gamma^\perp})\|_F^2 \\ &\leq 4 \cdot 64n^2\delta^2 + 64n^2\delta^2 \\ &= 5 \cdot 64n^2\delta^2 . \end{aligned} \tag{35}$$

Finally, using (24) and (35), the bound on the total error becomes

$$\begin{aligned}\|H\|_F^2 &= \|H^\Gamma\|_F^2 + \|H^{\Gamma^\perp}\|_F^2 \\ &\leq 2\delta^2 + 5 \cdot 64n^2\delta^2 \end{aligned} \tag{36}$$

which yields

$$\begin{aligned}\|X_0 - \hat{X}\|_F &= \|H\|_F \\ &\leq \sqrt{2\delta^2 + 5 \cdot 64n^2\delta^2} \qquad \text{(by (36))} \\ &\leq (\sqrt{2} + 8\sqrt{5}n)\delta , \end{aligned}$$

and we have shown that the error term is bounded linearly in δ as desired for Proposition A.5. ■

REFERENCES

[1] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1580791

[2] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, December 2009.

[3] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/090761793

[4] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011. [Online]. Available: http://doi.acm.org/10.1145/1970392.1970395

[5] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable Principal Component Pursuit," *ISIT 2010: Proceedings of IEEE International Symposium on Information Technology*, 2010. [Online]. Available: http://perception.csl.illinois.edu/matrix-rank/Files/isit_noise.pdf

[6] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *To be submitted*, December 2010.

[7] R. Mikut and M. Reischl, "Data mining tools," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 5, pp. 431–443, 2011. [Online]. Available: http://dx.doi.org/10.1002/widm.24

[8] D. Haughton, J. Deichmann, A. Eshghi, S. Sayek, N. Teebagy, and H. Topi, "A review of software packages for data mining," *The American Statistician*, vol. 57, no. 4, pp. 290–309, November 2003.

[9] I. Witten, E. Frank, M. Hall, and G. Holmes, *Data Mining: Practical Machine Learning Tools and Techniques*, ser. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011. [Online]. Available: http://books.google.com/books?id=bDtLM8CODsQC

[10] C. Bishop, *Pattern Recognition And Machine Learning*, ser. Information Science and Statistics. Springer, 2006. [Online]. Available: http://books.google.com/books?id=kTNoQgAACAAJ

[11] S. Marsland, *Machine Learning: An Algorithmic Perspective*, ser. Chapman & Hall/CRC machine learning & pattern recognition series. CRC Press, 2009. [Online]. Available: http://books.google.com/books?id=n66O8a4SWGEC

[12] S. Macskassy and F. Provost, "A brief survey of machine learning methods for classification in networked data and an application to suspicion scoring," in *Statistical Network Analysis: Models, Issues, and New Directions*, ser. Lecture Notes in Computer Science, E. Airoldi, D. Blei, S. Fienberg, A. Goldenberg, E. Xing, and A. Zheng, Eds. Springer Berlin / Heidelberg, 2007, vol. 4503, pp. 172–175, 10.1007/978-3-540-73133-7_13. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73133-7_13

[13] C. Krügel and T. Toth, "Distributed pattern detection for intrusion detection," in *Proc. of the Network and Distributed System Security Symp. (NDSS 2002)*, 2002. [Online]. Available: http://citeseer.ist.psu.edu/kruegel02distributed.html

[14] N. Boggs, S. Hiremagalore, A. Stavrou, and S. Stolfo, "Cross-domain collaborative anomaly detection: So far yet so close," in *Recent Advances in Intrusion Detection 14th International Symposium*, 2011.

[15] A. Ganesh, J. Wright, X. Li, E. Candes, and Y. Ma, "Dense error correction for low-rank matrices via principal component pursuit," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 1513 –1517.

[16] R. C. Paffenroth, P. C. Du Toit, L. L. Scharf, A. P. Jayasumana, V. W. Banadara, and R. Nong, "Space-time signal processing for distributed pattern detection in sensor networks," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 8393. SPIE, May 2012.

[17] R. C. Paffenroth, P. C. Du Toit, L. L. Scharf, A. P. Jayasumana, V. W. Banadara, and R. Nong, "Distributed pattern detection in cyber networks," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, I. V. Ternovskiy and P. Chin, Eds., vol. 8408, no. 1. SPIE, 2012, p. 84080J. [Online]. Available: http://link.aip.org/link/?PSI/8408/84080J/1

[18] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5454406

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Now Publishers, 2011. [Online]. Available: http://books.google.com/books?id=8MjgLpJ0_4YC

[20] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1137/080738970

[21] D. B. Parker, *Fighting computer crime: a new framework for protecting information*. New York, NY, USA: John Wiley & Sons, Inc., 1998.

[22] ——, "Toward a New Framework for Information Security," in *Computer Security Handbook*, 4th ed., S. Bosworth and M. E. Kabay, Eds. John Wiley and Sons, 2002, ch. 5.

[23] S. Boslaugh and P. Watters, *Statistics in a Nutshell: A Desktop Quick Reference*, ser. In a Nutshell. O'Reilly, 2008. [Online]. Available: http://books.google.com/books?id=ZnhgO65Pyl4C

[24] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936. [Online]. Available: http://dx.doi.org/10.1007/BF02288367

[25] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS 07*, vol. 35, no. 1, p. 109, 2007. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1254882.1254895

[26] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, , and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," University of Illinois Urbana-Champaign, Tech. Rep. UILU-ENG-09-2214, August 2009.

[27] "Internet 2 network," Web Page: http://noc.net.internet2.edu/i2network/index.html.

[28] P. J. Schreier and L. L. Scharf, *Statistical signal processing of complex-valued data: the theory of improper and noncircular signals*. Cambridge University Press, 2010.

[29] A. Aizerman, E. M. Braverman, and L. I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.

[30] R. Nelsen, *An Introduction to Copulas*, ser. Springer Series in Statistics. Springer, 2010. [Online]. Available: http://books.google.com/books?id=HhdjcgAACAAJ

[31] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011. [Online]. Available: http://www.sciencemag.org/content/334/6062/1518.abstract

[32] D. Gross, "Recovering low-rank matrices from few coefficients in any basis," *CoRR*, vol. abs/0910.1879, 2009.

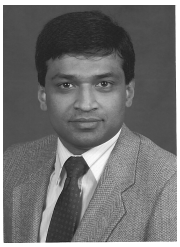
Randy Paffenroth Dr. Paffenroth graduated from Boston University with degrees in both mathematics and computer science. After defending his thesis in the spring of 1999, he was awarded his Ph.D. in Applied Mathematics from the University of Maryland in June of 1999. Following the attainment of his Ph.D., Dr. Paffenroth spent seven years as a Staff Scientist at the California Institute of Technology. In 2006 he joined the Numerica team and has since held the position of Computational Scientist and, most recently, Program Director. As Program Director, Dr. Paffenroth's responsibilities include the management of the software group at Numerica as well as process development, mathematical research, software engineering, and code design. As a Computational Scientist, he performed mathematical research and software development in tracking problems. Before joining the Numerica team, Dr. Paffenroth developed large software codes for high-order methods for boundary integral formulations of partial differential equations at the California Institute of Technology. His technical interests include the interaction between mathematics, computer science and software engineering.

Philip Du Toit Dr. Philip Du Toit joined the Numerica team as a research scientist after receiving the Ph.D. degree from the California Institute of Technology in Control and Dynamical Systems. He has previously received the Honors B.S. degree in Physics from Brigham Young University graduating *summa cum laude*, and an M.S. degree in Applied Mathematics from the University of Michigan. Dr. Du Toit has previously held the position of research scientist in the MRI Physics department of Cornell University Medical College. During his studies, he has been the recipient of several fellowships and awards: the Caltech Presidential Fellowship, an NSF fellowship at the University of Michigan, and is a BYU Edwin S. Hinckley Scholar.

Ryan Nong Dr. Nong graduated (with honor) from the California Institute of Technology with a B.Sc. in Applied and Computational Mathematics and a minor in Control and Dynamical Systems in 2005. He earned M.A. and Ph.D. degrees in Computational and Applied Mathematics from Rice University in 2008 and 2009, respectively. After spending eight months as a post-doctoral scholar working in the Department of Applied Mathematics and Applications at Sandia National Laboratories in Albuquerque, he joined Numerica in May of 2010. As a Computational Scientist, he is responsible for research and development of novel mathematical algorithms with applications to data fusion, image and signal processing, and other areas of interest to Numerica. His research interests include Numerical Linear Algebra, Numerica Analysis, Numerical Optimization and Large-Scale Scientific Computing.

LouisScharf Louis Scharf received his Ph.D. from the University of Washington, Seattle. He is currently Emeritus Professor of Electrical and Computer Engineering and Research Professor of Mathematics, with a joint appointment in Statistics, at Colorado State University. He has held visiting positions here and abroad, including Ecole Superieure d'Electricite, Ecole Superieure des Telecommunications, EURECOM, University of La Plata, Duke University, University of Wisconsin, and University of Tromso.

Professor Scharf's research interests are in statistical signal processing, as it applies to adaptive radar and sonar, electric power, cyber security, and wireless communication. He was Technical Program Chair for the 1980 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Tutorials Chair for ICASSP 2001; and Technical Program Chair for the Asilomar Conference on Signals, Systems, and Computers, 2002. He is past Chair of the Fellow Committee for the IEEE Signal Processing Society. He has received numerous awards for his contributions to statistical signal processing, including an IEEE Distinguished Lectureship; an IEEE Third Millennium Medal; the Technical Achievement Award from the IEEE Signal Processing Society (SPS); and the Society Award from the IEEE SPS. He is a Life Fellow of IEEE.



Anura Jayasumana Anura P. Jayasumana is a Professor in Electrical & Computer Engineering and Computer Science at Colorado State University. He is a member of the NSF Engineering Research Center for Collaborative Adaptive Sensing of Atmosphere (CASA). He received Ph.D. and M.S. in Electrical Engineering from Michigan State University, and B.Sc. in Electronic and Telecommunications Engineering with First Class Honors from University of Moratuwa, Sri Lanka. His current research interests include sensor networking, network protocols, and performance modeling. Anura also serves as a consultant to industry; his past clients range from startups to Fortune 100 companies. He is a member of Phi Kappa Phi, ACM and IEEE.

Vidarshana Bandara Vidarshana Bandara (M'05) received the B.Sc. degree in electrical and electronic engineering from University of Peradeniya, Sri Lanka, in 2005 and the M.Sc. degree in computer science from St. Cloud State University, Minnesota in 2008. Currently he is a Ph.D. student in electrical and computer engineering in Colorado State University, Colorado.