On Covariance Structure in Noisy, Big Data

Randy C. Paffenroth[a], Ryan Nong[a] and Philip C. Du Toit[a]

[a] Numerica Corporation, Loveland, CO, USA;

ABSTRACT

Herein we describe theory and algorithms for detecting covariance structures in large, noisy data sets. Our work uses ideas from matrix completion and robust principal component analysis to detect the presence of low-rank covariance matrices, even when the data is noisy, distorted by large corruptions, and only partially observed. In fact, the ability to handle partial observations combined with ideas from randomized algorithms for matrix decomposition enables us to produce asymptotically fast algorithms. Herein we will provide numerical demonstrations of the methods and their convergence properties. While such methods have applicability to many problems, including mathematical finance, crime analysis, and other large-scale sensor fusion problems, our inspiration arises from applying these methods in the context of cyber network intrusion detection.

1. INTRODUCTION

Correlation structure has a fundamental role to play in many data analysis problems. Pearson correlation matrices,[1] and their unnormalized cousins – covariance matrices, are a fundamental building block of many data fusion and target tracking problems when used to measure the correlation between various components of sensor measurements (e.g. see[2,3] and numerous references therein). Studying covariance matrices gives insight into many important attributes of sensors, such as "the relationship between uncertainty in range and uncertainty in angle." Herein it is our purpose to demonstrate how such ideas from the data fusion and target tracking "toolbox" can be used in other domains in interesting, and perhaps even surprising, ways.

In fact, recent works[4–6] demonstrate that second order structure, as encoded in correlation matrices, can be used in novel ways for analyzing computer networks. In particular, much insight can be derived by considering the *rank* of correlation matrices arising from large collections of cyber sensors such as port activities and packet rates. Such analysis has a long history, with the seminal paper of Eckart and Young[7] representing a foundational contribution and principle component analysis (PCA)[8] being in wide spread use in many fields.

As a simple example of how one might apply such analysis, suppose that one were to observe that a correlation matrix between a collection of sensors was *low-rank* (e.g. it had a number of zero eigenvalues), then one would be justified in concluding that this collection of sensors was *redundant*. In other words, the entirety of the sensor responses can be *predicted* by merely observing a *subset* of the sensor measurement. Similarly, an additional sensor which does not increase the rank of a correlation matrix *does not provide any additional information*. Its response is merely a linear combination of other sensor measurements.

One could also turn the above analysis "on its head" and wonder what happens if one were to remove a sensor, rather than adding one. If the rank were to remain fixed then one could conclude (in a sense that we will make precise later in the text) that the sensor did not add any new information, above and beyond what its brethren already provide. Many interesting questions immediately present themselves. Are there any general principles for understanding how much one might expect the rank of a covariance matrix to change based upon the action of a single sensor? How about a small collection of sensors? Can any insight be gained by noting that certain collections of sensors depart from these general principles? Would such sensors be, in some sense, anomalous?

Further author information: (Send correspondence to R.C.P.)
R.C.P.: E-mail: randy.paffenroth@numerica.us
R.N.: Email: ryan.nong@numerica.us
P.C.T.: E-mail: philip.dutoit@numerica.us

Precisely such questions have been addressed recently in the literature[4–6] and herein we aspire to make a contribution to this problem domain. In particular, for many interesting problems in computer network analysis the covariance matrices whose structure we wish to understand can be rather large, with problems sometimes having hundreds of thousands, if not millions of sensors. One merely needs to consider a moderate sized computer network (say with a thousand computers) with each computer measuring hundreds, if not thousands of quantities (e.g. CPU load, port activity, packet entropy, etc.) to comprehend the scale of the problem. Also, as we are interested in tackling problems of practical importance, one is forced to deal with the vagaries real-world data, including noise and outliers. Accordingly, our goal in this paper is to demonstrate methods for large scale analysis of covariance matrices that are robust to the issues of real world data. *It is not our intent here to present a full theoretical justification for our method but rather to provide their derivation and present suggestive empirical results.* Paths forward for theory of such methods can be found in[5] as well as in,[9–12] and it is our aim to provide a full theoretical justification in.[13]

We use the following notations throughout the paper. We reserve the letter L (and its various modifications such as L_0) to represent a low-rank matrix, with $\rho(L)$ being the rank of L and $\|L\|_*$ being the nuclear norm of L. In a similar vein, we reserve the letter S (and its various modifications such as S_0) to be a sparse matrix (i.e. one with few non-zero entries), with $\|S\|_0$ being the number of non-zero entries in S, and $\|S\|_1$ being the sum of the absolute values of the entries in S. Finally, we denote by $\mathcal{S}_{\bar{\epsilon}}(S)$ the point-wise shrinkage operator with $\bar{\epsilon}$ being the *matrix valued* threshold over the matrix S, i.e.,

$$\mathcal{S}_{\bar{\epsilon}}(S)_{i,j} = sign(S_{i,j}) \max(|S_{i,j}| - \bar{\epsilon}_{i,j}, 0).$$

2. LATENT SIGNAL MODELS

The subject of our exposition is determining when the rank of a large correlation matrix $M \in \mathbb{R}^{m\times m}$ can (perhaps drastically) be reduced by changing just a few of its entries. How might one proceed? Various naïve algorithms present themselves, such as brute force searches over the power set of all of the entries. Unfortunately, such an algorithm is clearly NP-hard in the size of the matrix m. Fortunately, there has been much work on various relaxations for this problem under the name Robust Principle Component Analysis (RPCA),[11,14,15] with a recent version, called eRPCA, being developed specifically for noisy problems.[5]

While we will certainly spill much ink in the derivation of such methods for noisy, big data problems, perhaps a more genial place to begin would be in a discussion of how covariance matrices might arise whose rank can be changed in a parsimonious fashion.

Accordingly, in this section we closely follow the notation and derivations found in.[4,5] In particular,[5] provides a detailed construction of our model, and here we will not reproduce the level of rigor presented there. Instead, herein we will merely provide sufficient high level intuition to lay down a foundation for our later exposition.

We begin by making the ansatz that our sensor measurements are encoded in a signal matrix, $Y \in \mathbb{R}^{m\times n}$, obeying a *latent time series model*. The key modeling principle is that the rows of Y each contain the measurements provided by a single sensor and that each sensor's response is, in fact, a linear combinations of underlying fundamental processes that are uncorrelated (or nearly uncorrelated). In particular, we assume that the raw data signal, Y, is composed as follows:

$$Y = AU + BV + N, \tag{1}$$

where $A \in \mathbb{R}^{m\times k}$ is dense but low-rank, $B \in \mathbb{R}^{m\times l}$ is sparse, and the matrices $U \in \mathbb{R}^{k\times n}$, $V \in \mathbb{R}^{l\times n}$, and $N \in \mathbb{R}^{m\times n}$ have mutually orthogonal rows.

In effect, our ansatz comprises the following structure:

AU: Since A is dense, each row in AU is a linear combination of *all* the underlying processes in U. Thus, U contains the (few) uncorrelated underlying processes that affect all the sensors. The ebb and flow of diurnal activity that affects many sensors, such as CPU load and network activity, is an example of such a process and would appear as a row in U;

BV: Since B is sparse, each row in BV is a linear combination of only a few of the underlying processes in V. Thus, V contains the uncorrelated processes that each simultaneously affect only a small subset of the sensors. Such processes may, or may not, exist for a particular scenario, but if they do exist then we wish to detect them;

N: N models the underlying processes that influence only individual sensors, and consequently does not represent any distributed behavior. These can be thought of as sensor *noise* which is uncorrelated with the environment.

As discussed in,[4,5] it is easy to show that *any* matrix $Y \in \mathbb{R}^{m\times n}$ can be decomposed as in (1) by computing its Singular Value Decomposition (SVD), $Y = \hat{U}\hat{\Sigma}\hat{V}^T$. One can then set $A = \hat{U}\hat{\Sigma}$, $U = \hat{V}^T$, and $B, V, N \equiv 0$ to produce the desired decomposition. In fact, given *any* desired B, V, and N, one can produce such a decomposition of Y using the SVD of $(Y - BV - N)$. Similarly, given *any* desired A, U, and N, one can also produce such a decomposition of Y by way of the SVD of $(Y - AU - N)$. What is more interesting, and not possible generically, is to produce a decomposition of Y where *simultaneously* A is low-rank *and* B is sparse and it is precisely such decompositions that RPCA[11,14,15] and eRPCA[5] seek to compute and can be used to detect anomalies in computer networks.[4,5]

Of course, the inclusion of noisy measurements may obscure the presence of an underlying pattern. The method we propose uses matrix completion and inequality constraints to explicitly account for uncertainty in the measurements, and thereby allow detection of patterns corrupted by noise. We require only that the streams of noise added to each trace are uncorrelated — there is no requirement on the size of the noise. (Note that if components (rows) of N are correlated, those components will appear in U or V).

3. CORRELATION MATRICES

As noted in,[5] one may be tempted to apply RPCA or eRPCA directly to the first order data matrix Y. Instead, for several reasons which we will delineate, we choose to examine second order correlation matrices. Accordingly, for ease of exposition and without loss of generality, we will presume throughout the text that Y is *normalized* so that

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

for some original raw data matrix $\tilde{Y}$ with row-wise empirical mean $\mu_{\tilde{Y}} \in \mathbb{R}^{m\times 1}$, and row-wise empirical 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. The above normalization is convenient since, for such a Y, the sample Pearson correlation matrix can be written as YY^T.[16]

As demonstrated in,[5] there are several advantages to analyzing the second order correlation matrix, $M = YY^T$, instead of the first order matrix Y. First, for many problems of interest, $m \ll n$ so that the matrix M is much smaller in size than the matrix Y. In other words, the number of measurements generated by a sensor is much larger than the number of sensors. This is advantageous in cyber domains[17,18] where it is infeasible to communicate the entire data matrix Y across the network. Second, studying M provides some measure of noise mitigation as compared to studying Y. As noted in,[5] 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.

Observe that our latent signal model in (1) leads to a decomposition of M as

$$M = YY^T = A\Sigma_{UU}A^T + B\Sigma_{VV}B^T + \Sigma_{NN} + W, \tag{2}$$

for diagonal matrices Σ_{UU}, Σ_{VV}, and Σ_{NN}, and where W is an error matrix that coarsely accounts for modeling errors. Note that the cross terms in M drop out because of the orthogonality assumptions on U, V, and N. By

setting $L := A\Sigma_{UU}A^T$, $S := B\Sigma_{VV}B^T$, and $E := \Sigma_{NN} + W$, we can now make our notation consistent with the RPCA literature[11] and write,

$$M = YY^T = L + S + E\,. \tag{3}$$

Note how the decomposition of correlation matrix M implies that Y has interesting structure. That is, the existence of low rank L in the decomposition of YY^T implies the existence of a low rank A, and the existence of a sparse S implies the existence of a sparse B^*. In particular, because of noise, a correlation matrix M will, generically, have full rank m. It is easy to show that any such matrix can have its rank reduced to $m - \sqrt{s}$ by changing s of its entries, but full rank matrices whose rank can be reduced to $m - \sqrt{s}$ by changing *less than* s elements are rare † and their structure has been demonstrated to lead to inferences about computer networks.[5]

4. PROBLEM SETUP

In the prototypical RPCA problem, we are given a matrix M, such as discussed in the previous section, that is formed by

$$M - L_t - S_t = 0, \tag{4}$$

where the true L_t is low-rank, the true S_t is sparse, and we are asked to recover L_t and S_t via computing their corresponding approximations L and S. Of course, such methods can be used to treat more general matrices than just correlation matrices, but correlation matrices are our focus here.

This decomposition is computed by relaxing an intractable minimization using the rank of L and the sparsity of S

$$\begin{aligned} L, S = \arg\min_{L_0, S_0} \rho(L_0) + \lambda\|S_0\|_0, \\ \text{s.t. } M - L_0 - S_0 = 0, \end{aligned} \tag{5}$$

to a *convex* minimization involving the nuclear norm of L and the 1-norm of S

$$\begin{aligned} L, S = \arg\min_{L_0, S_0} \|L_0\|_* + \lambda\|S_0\|_1, \\ \text{s.t. } M - L_0 - S_0 = 0 \end{aligned} \tag{6}$$

(see[11] for details).

In a similar vein, the prototypical eRPCA problem relaxes (4) by merely requiring that

$$|M - L_0 - S_0| \preceq \bar{\epsilon}, \tag{7}$$

where $\bar{\epsilon} \in \mathbb{R}^{m\times m}$ is a *matrix* of error bounds and $\preceq$ represents *point-wise* less than. (7) can then be plugged into the relaxed optimization (6) to get

*The converse of this statement is substantially more delicate. For example a "sparse" B with all of its entries in one column can lead to a "dense" $S = B\Sigma_{VV}B^T$. This idea is closely related to incoherence measures on sparse matrices S.[19]

†In fact, they are measure 0 under any reasonable probability distribution on the entries of M.

$$L, S = \arg\min_{L_0, S_0} \|L_0\|_* + \lambda \|S_0\|_1, \quad \text{s.t. } |M - L_0 - S_0| \preceq \bar{\epsilon}, \tag{8}$$

and its equivalent formulation

$$L_1, S_1 = \arg\min_{L_0, S_0} \|L_0\|_* + \lambda \|\mathcal{S}_{\bar{\epsilon}}(S_0)\|_1, \quad \text{s.t. } M - L_0 - S_0 = 0, \; L, S = L_1, \mathcal{S}_{\bar{\epsilon}}(S_1). \tag{9}$$

(see[5] for details). The first formulation is often used for constructing theoretical arguments while the later formulation is used for numerical calculations. In,[5] it is proven that the two formulations are equivalent (i.e. an L, S pair minimizes one if, and only if, it minimizes the other) so herein we use them interchangeably.

Let us now present the following nomenclature:

- An entry M_{ij} of M is called **observed** if $\bar{\epsilon}_{ij} = 0$.
- An entry M_{ij} of M is called **partially observed** (or **bounded**) if $0 < \bar{\epsilon}_{ij} < \infty$.
- An entry M_{ij} of M is called **unobserved** if $\bar{\epsilon}_{ij} = \infty$ (or equivalently if $\bar{\epsilon}_{ij}$ is arbitrarily large).

Note that our use of the term "unobserved" is intended to mirror the use of that term in the "Matrix Completion" literature.[14,20,21] Here we generalize those ideas slightly by treating the case where one has partial information about some of the entries in M. In other words, an entry M_{ij} may not be known precisely, but one is armed with a bound on the possible values it can take.

With the above notation in mind, we are now ready to state our main empirical result.

Empirical Result 4.1. *Suppose that one is given a (correlation) matrix M with $\mathcal{O}(m)$ entries that are either observed or partially observed (the remainder of the entries being unobserved) and a bound l on the rank of L which is independent of the size m of M. Then each iteration in the optimization problem* (9) *costs $\mathcal{O}(m)$ in both time and memory.*

In other words, for each iteration, *the cost of the minimization is a function of the number of observed entries in M rather than the number of total entries in M.* Do such matrices arise in practice? Indeed, the idea being that, for each sensor (i.e. row of Y) we only observe correlations with a fixed number of other sensors. Thinking about a network, one could presume that as the network grows, the number of neighbors of each node is fixed. E.g. adding a router at one end of a network doesn't change the number of neighbors for nodes at the other end of the network. Accordingly, one can generate a correlation matrix M of the desired form by merely computing the correlations between neighboring nodes in the network.

Two technical details bear a small measure of discussion at this point, with a more fulsome explanation to be found in.[5] First, as we see from our latent signal model for correlation matrices M in (2), the uncorrelated sensor noise appears as a diagonal matrix E, which, if not treated appropriately, will tend to make the correlation matrix M full rank. Fortunately, this is easily mitigated in our formulation by making the diagonal entries $\bar{\epsilon}_{ii}$ of the error bound matrix $\bar{\epsilon}$ arbitrarily large, so that E does not appear in our decomposition. Second, any entry S_{ij} of S which has the property that $\bar{\epsilon}_{ij} > S_{ij}$ will not appear as a non-zero entry of $S_{0_{ij}}$.

5. ALGORITHM DERIVATION

To solve the optimization problem in (9) with a fast method that has our desired asymptotic performance properties we use an Augmented Lagrange Multiplier (ALM) formulation which we solve using the Alternating Direction Method of Multipliers (ADMM).[22]

To wit, we first observe that our constrained optimization problem can we rewritten, in standard fashion,[22] as an *unconstrained* minimization problem by relaxing the constraints using an ALM as

$$\mathcal{L}(L, S, Y, \mu) := \|L\|_* + \lambda\|\mathcal{S}_{\bar{\epsilon}}(S)\|_1 + \langle Y, M - L - S\rangle + \frac{\mu}{2}\langle M - L - S, M - L - S\rangle, \tag{10}$$

where Y is the Lagrange multiplier for the constraint $M - L_0 - S_0 = 0$ and μ is the coupling constant for the augmented Lagrangian term $\langle M - L_0 - S_0, M - L_0 - S_0\rangle$ for the same constraint. By collecting terms one can see that (10) is equivalent to

$$\mathcal{L}(L, S, Y, \mu) = \|L\|_* + \lambda\|\mathcal{S}_{\bar{\epsilon}}(S)\|_1 + \frac{\mu}{2}\left\langle M - L - S + \frac{1}{\mu}Y, M - L - S + \frac{1}{\mu}Y\right\rangle - \frac{1}{2\mu}\langle Y, Y\rangle, \tag{11}$$

which, after applying the definition of the Frobenius norm, can be written in a standard form as

$$\mathcal{L}(L, S, Y, \mu) = \|L\|_* + \lambda\|\mathcal{S}_{\bar{\epsilon}}(S)\|_1 + \frac{\mu}{2}\|M - L - S + \frac{1}{\mu}Y\|_F^2 - \frac{1}{2\mu}\|Y\|_F^2 \tag{12}$$

just as in.[5]

Here we depart from[5] in that we wish to construct an algorithm which requires an $\mathcal{O}(n)$ computational cost, while evaluating the Lagrangian in (12), on the face of it, requires $\mathcal{O}(n^2)$. Accordingly, we must rewrite the Lagrangian in a different form.

Drawing inspiration from the Matrix Completion literature[14,20,21] we define Ω to be the set of observed entries in M and define the projection operator P_Ω as the projection onto this set. Analogously, let $\bar{\Omega}$ be the set of unobserved entries in M and define the projection operator $P_{\bar{\Omega}}$ as the projection onto this set. It is readily observed from this definition that

$$A = P_\Omega(A) + P_{\bar{\Omega}}(A).$$

Accordingly, we write our new problem definition as

$$\begin{aligned} L_1, S_1 = \arg\min_{L_0, S_0} \|L_0\|_* + \lambda\|\mathcal{S}_{\bar{\epsilon}}(P_\Omega(S_0))\|_1 \\ \text{s.t. } P_\Omega(M - L_0 - S_0) = 0, \\ L, S = L_1, \mathcal{S}_{\bar{\epsilon}}(P_\Omega(S_1)), \end{aligned} \tag{13}$$

which gives a Lagrangian (after completing the square and various manipulations) as follows:

$$\mathcal{L}(L, S, Y, \mu) = \|L\|_* + \lambda\|\mathcal{S}_{\bar{\epsilon}}(P_\Omega(S))\|_1 + \frac{\mu}{2}\|P_\Omega(M - L - S) + \frac{1}{\mu}Y\|_F^2 - \frac{1}{2\mu}\|Y\|_F^2. \tag{14}$$

Empirically, the minimizers of problem (12) and (14) agree to high precision. The proof of their equivalence is not complete, but the methods used to prove the equivalence of (8) and (9) in[5] will likely provide a good starting point. The key idea that suggests their equivalence is to observe that $P_{\Omega_{ij}}$ and large error bounds $\bar{\epsilon}_{ij}$ play precisely the same role. They both allow L_{ij} to take whatever value which minimizes the nuclear norm $\|L\|_*$ while simultaneously allowing $S_{ij} = 0$. Accordingly, one would merely need to prove that a minimizer to one problem provides a minimizer to the other for the observed and partially observed entries, which is accomplished in[5] using a proof by contradiction.

What have we gained? Note that the terms

$$\lambda\|\mathcal{S}_{\bar{\epsilon}}(S)\|_1 + \frac{\mu}{2}\|M - L - S + \frac{1}{\mu}Y\|_F^2 - \frac{1}{2\mu}\|Y\|_F^2 \tag{15}$$

in the Lagrangian in (12) depend on all the entries of M, even those for which $\bar{\epsilon}_{ij}$ is arbitrarily large, while the analogous terms in the Lagrangian in (14), shown below,

$$\lambda\|\mathcal{S}_{\bar{\epsilon}}(P_\Omega(S))\|_1 + \frac{\mu}{2}\|P_\Omega(M - L - S) + \frac{1}{\mu}Y\|_F^2 - \frac{1}{2\mu}\|Y\|_F^2 \tag{16}$$

do *not depend on the unobserved entries* in $P_{\bar{\Omega}}$. Accordingly, the above terms can be evaluated in (14) using $\mathcal{O}(m)$ operations, as long as the number of observed and partially observed entries in P_Ω is also of $\mathcal{O}(m)$.

The only term that remains for our consideration in (14) is $\|L\|_*$. One might rightly wonder how to get the $\mathcal{O}(m)$ storage and computation bounds that we desire, since our recovered L is a dense matrix and has m^2 entries. The key insight is that the algorithm never needs to store $L \in \mathbb{R}^{m\times m}$ explicitly if we require the user to input an $l \ll m$ (independent of m) which bounds the maximum rank of the desired L. With this, L can be stored (and later returned) by its SVD $L = U\Sigma V^T$ where $U \in \mathbb{R}^{m\times l}$, $\Sigma \in \mathbb{R}^{l\times l}$, and $V^T \in \mathbb{R}^{l\times m}$. Clearly, storing L as $U\Sigma V^T$ requires only $\mathcal{O}(m)$ storage and evaluating individual elements of L, as required in (16), can be done in $\mathcal{O}(k^2)$ operations, which is independent of m.

Accordingly, following this line of reasoning, one can see that the input to the algorithm requires $\mathcal{O}(m)$ storage for $P_\Omega(M)$ and $P_\Omega(\bar{\epsilon})$; and the internal storage and output storage are also $\mathcal{O}(m)$ for $P_\Omega(M)$, $P_\Omega(\bar{\epsilon})$, $P_\Omega(S)$, U, Σ, and V.

6. IMPLEMENTATION DETAILS

As is classic with ADMM methods, we wish to alternate minimizing the Lagrangian $\mathcal{L}$ in (14) with respect to L, with S fixed; minimizing $\mathcal{L}$ with respect to S, with L fixed; and updating the Lagrange multiplier. After dropping constant terms, this algorithm can be written as

$$L_{k+1} = \arg\min_L \|L\|_* + \frac{\mu}{2}\|P_\Omega(M - L - S_k) + \frac{1}{\mu}Y_k\|_F^2 \tag{17}$$

$$S_{k+1} = \arg\min_S \|\mathcal{S}_\epsilon(P_\Omega(S))\|_1 + \frac{\mu}{2}\|P_\Omega(M - L_{k+1} - S) + \frac{1}{\mu}Y_k\|_F^2 \tag{18}$$

$$Y_{k+1} = Y_k + \mu(M - L_{k+1} - S_{k+1}). \tag{19}$$

The minimum of $\mathcal{L}$ with respect to S with L fixed, as in (18) is quite straight forward and the majority of the details can be found in[5] with a fulsome treatment to appear in.[13]

Accordingly, let us focus our attention on the much more delicate minimum of $\mathcal{L}$ with respect to L with S fixed, as in (17). Let us assume that we are at iteration k, having already computed L_k, S_k, and Y_k, and wish to compute L_{k+1}. After multiplying through by $\frac{1}{\mu}$, our minimization can be written as

$$L_{k+1} = \arg\min_L \frac{1}{2}\|P_\Omega(M - L - S_k) + \frac{1}{\mu}Y_k\|_F^2 + \frac{1}{\mu}\|L\|_*,$$

which is equivalent to

$$L_{k+1} = \arg\min_L \frac{1}{2}\|P_\Omega(M - S_k) - P_\Omega(L) + \frac{1}{\mu}Y_k\|_F^2 + \frac{1}{\mu}\|L\|_*.$$

Adding $P_{\bar{\Omega}}(L) - P_{\bar{\Omega}}(L) = 0$ to get

$$L_{k+1} = \arg\min_L \frac{1}{2}\|P_{\bar{\Omega}}(L) - P_{\bar{\Omega}}(L) + P_\Omega(M - S_k) - P_\Omega(L) + \frac{1}{\mu}Y_k\|_F^2 + \frac{1}{\mu}\|L\|_*.$$

Combining up terms gives

$$L_{k+1} = \arg\min_L \frac{1}{2}\|P_{\bar{\Omega}}(L) + P_\Omega(M - S_k) - L + \frac{1}{\mu}Y_k\|_F^2 + \frac{1}{\mu}\|L\|_*.$$

Finally, after multiplying through by -1 in the first term gives us the desired form:

$$L_{k+1} = \arg\min_L \frac{1}{2}\|L - \left(P_{\bar{\Omega}}(L) + P_\Omega(M - S_k) + \frac{1}{\mu}Y_k\right)\|_F^2 + \frac{1}{\mu}\|L\|_*. \tag{20}$$

Now, consider the following result, which can be found as Theorem 2.1 on pg 5 of,[10]

THEOREM 6.1. *For each $\tau \geq 0$ and $A \in \mathbb{R}^{n\times m}$, the singular value shrinkage operator $\mathcal{D}_\tau$ defined by*

$$\mathcal{D}_\tau(A) = \mathcal{D}_\tau(U\Sigma V^T) = U\mathcal{S}_\tau(\Sigma)V^T$$

obeys

$$\mathcal{D}_\tau(A) = \arg\min_B \left\{\frac{1}{2}\|B - A\|_F^2 + \tau\|B\|_*\right\}.$$

Based on this result, the following corollary follows:

COROLLARY 6.1. *For each $\tau \geq 0$ and $A \in \mathbb{R}^{n\times m}$, the singular value shrinkage operator $\mathcal{D}_\tau$ defined by*

$$\mathcal{D}_\tau(A) = \mathcal{D}_\tau(U\Sigma V^T) = U\mathcal{S}_\tau(\Sigma)V^T$$

obeys

$$C + \mathcal{D}_\tau(A) = \arg\min_B \left\{\frac{1}{2}\|B - C - A\|_F^2 + \tau\|B - C\|_*\right\}.$$

Proof. Starting with

$$\arg\min_B \left\{\frac{1}{2}\|B - A\|_F^2 + \tau\|B\|_*\right\}$$

we can add $C - C = 0$ twice to get

$$\arg\min_B \left\{\frac{1}{2}\|B + C - C - A\|_F^2 + \tau\|B + C - C\|_*\right\}$$

We can now introduce a change of variables $\hat{B} = B + C$ to get

$$\arg\min_{\hat{B}-C} \left\{\frac{1}{2}\|\hat{B} - C - A\|_F^2 + \tau\|\hat{B} - C\|_*\right\},$$

which, with constant C, is also

$$\arg\min_{\hat{B}} \left\{\frac{1}{2}\|\hat{B} - C - A\|_F^2 + \tau\|\hat{B} - C\|_*\right\}$$

which is the desired result. □

Back to (20), our implementation approximates the L in $P_{\bar{\Omega}}(L)$ by using the value of L *from the previous iteration* L_k while searching for a new optimal value of L_{k+1} to minimize $\frac{1}{2}\|L-\left(P_{\bar{\Omega}}(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right)\|_F^2+\frac{1}{\mu}\|L\|_*$. This approximation leads to an *inexact ADMM*,[23,24] whose convergence we study in Section 7.

Then, by applying Corollary 6.1 to (20), we can see that

$$L_{k+1}=\arg\min_L \frac{1}{2}\|L-\left(P_{\bar{\Omega}}(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right)\|_F^2+\frac{1}{\mu}\|L\|_*=\mathcal{D}_{\frac{1}{\mu}}\left(P_{\bar{\Omega}}(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right).$$

One merely adds $P_\Omega(L_k)-P_\Omega(L_k)=0$ to the argument of $\mathcal{D}_{\frac{1}{\mu}}$, and collects terms, to obtain

$$\mathcal{D}_{\frac{1}{\mu}}\left(P_{\bar{\Omega}}(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right)=\mathcal{D}_{\frac{1}{\mu}}\left(L_k-P_\Omega(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right).$$

Note that $(L-P_\Omega(L)+P_\Omega(M-S)+\frac{1}{\mu}Y$ can be applied in $\mathcal{O}(m)$ time to a vector x as long as L is represented by an SVD, L has rank l independent of m, and $P_\Omega(L)+P_\Omega(M-S)+\frac{1}{\mu}Y$ has only $\mathcal{O}(m)$ entries, by observing that

$$\left(L_k-P_\Omega(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right)x= \tag{21}$$

$$L_kx-\left(P_\Omega(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right)x= \tag{22}$$

$$(U_k\Sigma_kV_k^T)x-\left(P_\Omega(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right)x= \tag{23}$$

$$U_k(\Sigma_k(V_k^Tx))-\left(P_\Omega(L_k)+P_\Omega(M-S_k)+\frac{1}{\mu}Y_k\right)x. \tag{24}$$

The requisite SVD to evaluate $\mathcal{D}_{\frac{1}{\mu}}$ can now be constructed using a *randomized* SVD algorithm[12] which requires l applications of the above operator. Therefore the total cost of the SVD is $\mathcal{O}(ml)$ operations and we get our desired performance bound per iteration of the ADMM method.

With this, we want to reaffirm our claim above that the computational cost of each iteration in the optimization problem (9) is $\mathcal{O}(m)$. In terms of the total computational cost of the algorithm, as shown below in Figure 3, our empirical numerical results suggest a sub-linear growth in the number of iterations.

7. PERFORMANCE RESULTS

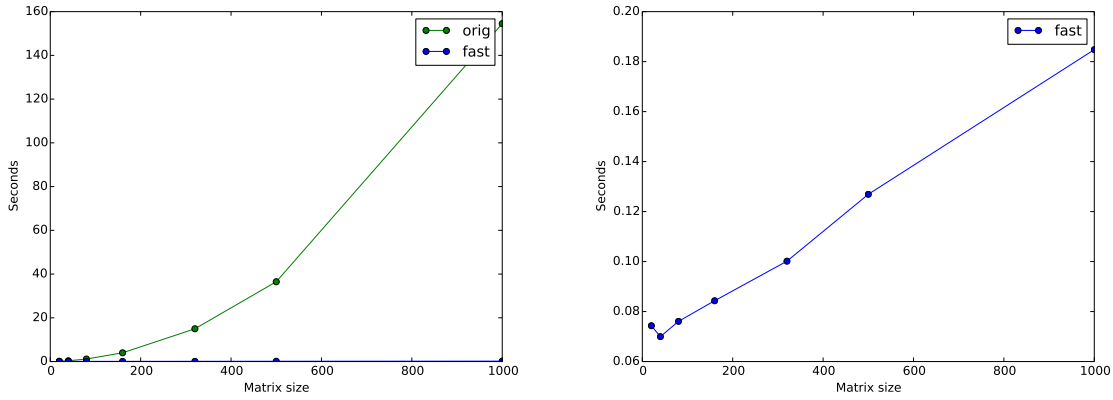
To test our algorithms, we construct matrices of the following form. We begin by fixing a rank k, with all of the tests in this document using $k=2$. Then, in keeping with our latent signal model we construct a matrix $A\in\mathbb{R}^{m\times k}$ with each entry drawn from a zero-mean unit-variance normal distribution $N(0,1)$. We then construct a symmetric low-rank matrix M as $M=AA^T$ and make $10m$ uniformly distributed random observations of the entries of M. For each observed entry M_{ij} in M we use an error tolerance $\bar{\epsilon}_{ij}=0.1M_{ij}$ and, to simulate the effect of sparse anomalies, each observed entry of M is perturbed by adding 1 with probabilty 0.02. The coupling parameters λ and ρ are set to be $\lambda=\frac{1}{\sqrt{10}}$ (in accordance with[21]) and $\rho=1.05$ (based upon empirical experimentation and roughly in accordance with[5]). Finally, we choose a convergence criterion as

$$\frac{\|M-L_0-S_0\|_F}{\|M\|_F}<10^{-5}. \tag{25}$$

As mentioned above, the computational cost of our new algorithm is $\mathcal{O}(n)$, per iteration. In the following, we present examples to demonstrate that this is in fact observed numerically.

In the first example, we run both the algorithms for problems of size up to 1000 and compare the time required to return the answers. The comptutional cost of the two algorithms are recorded in Figure 1.

In the second example, for the new algorithm, we increase the size of the problem to $100,000$ where it takes merely 20 seconds to return the answer. The computational cost appears linear as shown in Figure 2.



(a) Fast and slow algorithms

(b) Just fast algorithm

Figure 1. Problems of different sizes up to 1000 are considered. In (a) we plot the time required to return the results of both algorithms. In (b) we only show the time required to return the results with respect to size m and the computational cost does appear $\mathcal{O}(m)$.

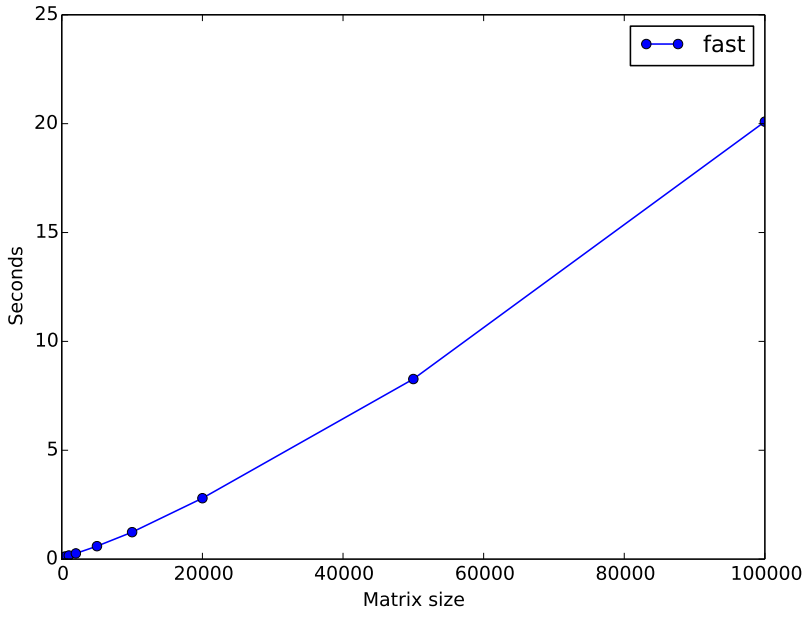


Figure 2. Problems of different sizes up to $100,000$ are considered. It takes barely 20 seconds to return the result. And the computational cost does appear linear.

Now, in order for the algorithm to be $\mathcal{O}(m)$ computationally, the number of iterations must be independent

of m. To test this claim, we also perform a test to be assured that this is in fact the case. In Figure 3, we present the results of iteration count as m increases. Empirically speaking, it appears that the number of iterations grows sub-linearly.

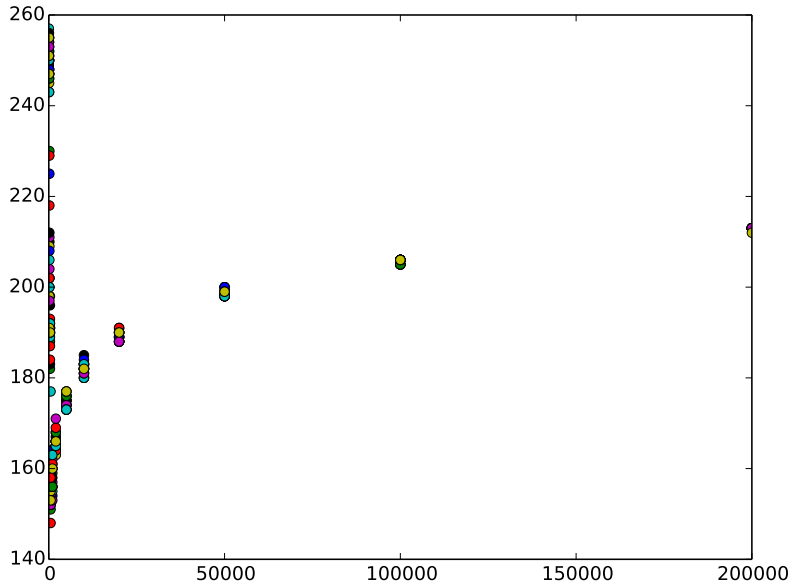


Figure 3. We record iteration counts as m increases. For each matrix size we perform twenty Monte-Carlo iterations and plot the number of iterations required for convergence. For small matrices, the number of iterations can vary substantially, but for large matrices the number of iterations required for convergence are tightly clustered. From an empirical perspective, it appears that the number of required iterations grows sub-linearly.

Note that in current literature,[23] the convergence of an *inexact ADMM* method, such as we use, is not guaranteed theoretically. However, convergence has been widely observed numerically.[24] In the following, we also perform a few experiments to test the convergence of the error and the Lagrangian. Note that in order to guarantee a *total* $\mathcal{O}(m)$ computational cost for our new algorithm, the iteration count has to be independent of m, which we do not claim here. Rather, we merely note that the growth appears to be sub-linear based upon numerical experiments.

8. CONCLUSIONS

We have presented theory and algorithms for detecting covariance structures in large, noisy data sets. In order to detect the presence of low-rank covariance matrices, our methods are based upon theories in compressed sensing, matrix completion and robust principal component analysis. This can be achieved even when the data is noisy, distorted by large corruptions, and only partially observed. The ability to handle partial observations combined with ideas from randomized algorithms for matrix decomposition enables us to produce asymptotically fast algorithms. Specifically, we have shown and demonstrated that large-scale low-rank covariance matrices can be recovered very quickly provided that the input data from sensors is large and sparse. While such methods have applicability to many problems, including mathematical finance, crime analysis, and other large-scale sensor fusion problems, our inspiration arises from applying these methods in the context of cyber network intrusion detection.

9. ACKNOWLEDGMENTS

All authors gratefully acknowledge funding for this research from the Air Force Office of Scientific Research under STTR contract FA9550-10-C-0090 as well as Numerica Corporation for their IRAD support. We would

also like to acknowledge our collaborators Dr. Anura Jayasumana, Dr. Louis Scharf, and Vidarshana Bandara at Colorado State University for many wonderful conversations.

REFERENCES

[1] Bishop, C., [*Pattern recognition and machine learning*], Springer New York. (2006).

[2] Blackman, S. and Popoli, R., [*Design and analysis of modern tracking systems*], Artech House, Boston, London (1999).

[3] Bar-Shalom, Y. and Li, X. R., [*Multitarget Multisensor Tracking: Principles and Techniques*], YBS Publishing, Storrs, CT (1995).

[4] Paffenroth, R., Toit, P. D., Scharf, L., Jayasumana, A. P., Bandara, V., and Nong, R., "Distributed pattern detection in cyber networks," in [*Cyber Sensing*], **8393** (2012).

[5] Paffenroth, R., Toit, P. D., Nong, R., Scharf, L. L., Jayasumana, A., and Bandara, V., "Space-time signal processing for distributed pattern detection in sensor networks," *IEEE Journal of Selected Topics in Signal Processing* **6**(1) (2013).

[6] Paffenroth, R., Toit, P. D., Scharf, L., and Jayasumana, A., "Space-Time Signal Processing for Detecting and Classifying Distributed Attacks in Networks," in [*2011 Joint Meeting of the Military Sensing Symposia (MSS) Specialty Groups*], (2011).

[7] Eckart, C. and Young, G., "The approximation of one matrix by another of lower rank," *Psychometrika* **1**, 211–218 (1936).

[8] I T, J., [*"Principal Component Analysis,2nd ed"*], vol. 98 (2002).

[9] Lin, Z., Chen, M., Wu, L., and Ma, Y., "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," Tech. Rep. ENG-09-2215, UIUC (Nov. 2009).

[10] Cai, J., Candès, E., and Shen, Z., "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization* , 1–28 (2010).

[11] Candés, E. J., Li, X., Ma, Y., Wright, J., and Candes, E. J., "Robust Principal Component Analysis?," *Submitted for publication* **58**(3), 1–37 (2010).

[12] Halko, N., Martinsson, P., and Tropp, J., "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *SIAM review* **53**(2), 217–288 (2011).

[13] Paffenroth, R., Nong, R., and Toit, P. D., "Fast Alternating Direction Method of Multipliers Solvers for Robust Principal Component Analysis and Big Data," *in preparation* (2013).

[14] Chen, Y., Xu, H., Caramanis, C., and Sanghavi, S., "Robust Matrix Completion with Corrupted Columns," 32 (Feb. 2011).

[15] Xu, H. and Caramanis, C., "Robust PCA via outlier pursuit," *Information Theory, IEEE* , 1–22 (2010).

[16] Boslaugh, S. and Watters, D. P. A., [*Statistics in a Nutshell: A Desktop Quick Reference (Google eBook)*], O'Reilly Media, Inc. (2009).

[17] Parker, D. B., [*Fighting Computer Crime*], John Wiley & Sons (1998).

[18] Parker, D., "Toward a New Framework for Information Security," in [*The Computer Security Handbook*], John Wiley & Sons (2002).

[19] Candès, E. and Romberg, J., "Sparsity and Incoherence in Compressive Sampling," *Inverse Problems* **23**, 969–986 (June 2007).

[20] Candès, E. and Recht, B., "Exact matrix completion via convex optimization," *To appear in Found. of Comput. Math.* (2009).

[21] Candes, E. J. and Plan, Y., "Matrix Completion With Noise," *Proceedings of the IEEE* **98**(6), 11 (2009).

[22] Boyd, S., "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning* **3**(1), 1–122 (2010).

[23] Lin, Z., Chen, M., and Ma, Y., "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," *arXiv:1009.5055v2 [math.OC]* (2011).

[24] Ng, M., Wang, F., and Yuan, X., "Inexact Alternating Direction Methods for Image Recovery," *SIAM J. Sci. Comput.* (2011).