

RANDOMIZED PROJECTION METHODS FOR LINEAR SYSTEMS WITH ARBITRARILY LARGE SPARSE CORRUPTIONS*

JAMIE HADDOCK[†] AND DEANNA NEEDELL[†]

Abstract. In applications like medical imaging, error correction, and sensor networks, one needs to solve large-scale linear systems that may be corrupted by a small number of arbitrarily large corruptions. We consider solving such large-scale systems of linear equations $A\mathbf{x} = \mathbf{b}$ that are inconsistent due to corruptions in the measurement vector \mathbf{b} . With this as our motivating example, we develop an approach for this setting that allows detection of the corrupted entries and thus convergence to the “true” solution of the original system. We provide analytical justification for our approaches as well as experimental evidence on real and synthetic systems.

Key words. corrupted linear systems, iterative methods, randomized projection methods

AMS subject classifications. 65F10, 65F20, 65F22

DOI. 10.1137/18M1179213

1. Introduction. We consider solving large-scale systems of linear equations represented by a matrix $A \in \mathbb{R}^{m \times n}$ and vector $\mathbf{b} \in \mathbb{R}^m$. We are interested in the highly overdetermined setting, where $m \gg n$, which means the system need not necessarily have a solution. One may then seek the least squares solution \mathbf{x}_{LS} which minimizes $\|A\mathbf{x} - \mathbf{b}\|^2$ (where $\|\cdot\|$ denotes the Euclidean norm); many efficient solvers have been developed that converge to such a solution. An alternative setting is one where there is a solution \mathbf{x}^* (which we refer to as the *pseudosolution*) to our desired system $A\mathbf{x} = \mathbf{b}^*$, but rather than observing \mathbf{b}^* we only have access to a corrupted version, \mathbf{b} , where $\mathbf{b} = \mathbf{b}^* + \mathbf{b}_C$. When the number of nonzero entries in \mathbf{b}_C , denoted $\|\mathbf{b}_C\|_0$, is small relative to m , one may still hope to recover the “true” solution \mathbf{x}^* .¹ This type of sparse corruption models many applications, ranging from medical imaging to sensor networks and error correcting codes. For example, a small number of sensors may malfunction, resulting in large catastrophic reporting errors in the vector \mathbf{b} ; since the reporting errors themselves may be arbitrarily large, the least squares solution is far from the desired solution, but since the number of such reporting errors is small, we may still hope to recover the true solution to the uncorrupted system. We emphasize that such a pseudosolution \mathbf{x}^* may be very far from the least squares solution \mathbf{x}_{LS} when the entries in \mathbf{b}_C are large, even when there are only a few nonzero corruptions; see Figure 1 for a visual. Similar types of sparse errors may also appear in medical imaging from artifacts or system malfunctions, or in error correcting codes from transmission errors. Indeed, the problem of so-called *sparse recovery* is well studied in the approximation and compressed sensing literature [FR13, EK12].

*Received by the editors April 9, 2018; accepted for publication (in revised form) December 18, 2018; published electronically October 29, 2019.

<https://doi.org/10.1137/18M1179213>

Funding: This material is based upon work supported by the National Science Foundation under grant DMS-1440140 while the authors were in residence at the Mathematical Sciences Research Institute in Berkeley, California, during the Fall 2017 semester. The first author was also partially supported by NSF grant DMS-1522158 and the University of California, Davis Dissertation Fellowship. The second author was also supported by NSF CAREER award 1348721 and NSF BIGDATA 1740325.

[†]Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095 (jhaddock@math.ucdavis.edu, deanna@math.ucla.edu).

¹This paper extends work previously presented in [HN17].

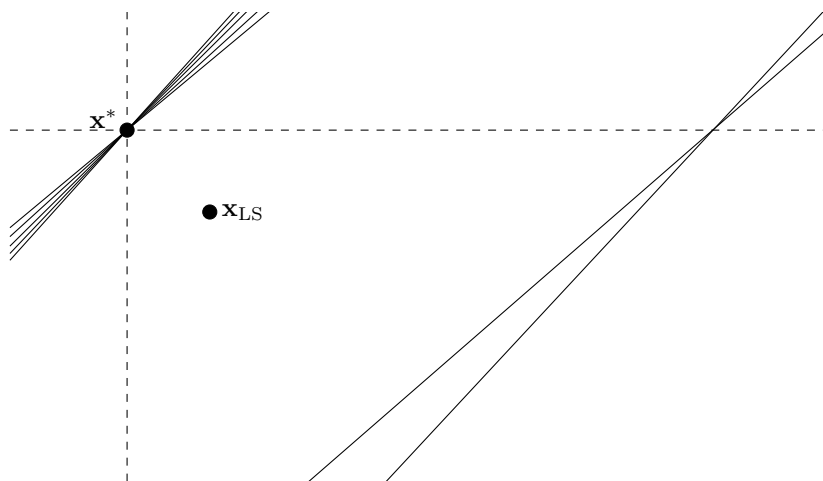


FIG. 1. A system for which the pseudosolution \mathbf{x}^* is very far from the least squares solution \mathbf{x}_{LS} . Lines represent the hyperplanes consisting of all systems $\{\mathbf{x} : \mathbf{a}_i^T \mathbf{x} = b_i\}$ for rows \mathbf{a}_i^T of A .

However, in this paper we are concerned with the setting where the system is highly *overdetermined*, the errors in \mathbf{b} are sparse and large, and the system may be so large-scale that it cannot be fully loaded into memory. This latter property has sparked a recent resurgence of work in the area of iterative solvers that do not need access to the entire system at once [GHJ75, HLL78, Nat01, SV09]. Our work is motivated by such iterative methods.

It is important to point out that solving for the pseudosolution of systems $A\mathbf{x} = \mathbf{b} = \mathbf{b}^* + \mathbf{b}_C$, where $\|\mathbf{b}_C\|_0$ is small relative to m is related to finding a solution of a large consistent system within an inconsistent system. The problem of finding the maximal consistent subsystem of an inconsistent system is known as MAX-FS and it is known to be NP-hard without a polynomial-time approximation scheme (PTAS) [AK95]. This problem is one of the focuses of infeasibility analysis, the study of changes necessary to make an infeasible system of linear constraints feasible [MKC00]. There are approximation algorithms [NR08] or, of course, one can solve the problem in a brute force manner. Approaches for solving this problem generally fall into two categories, heuristic methods which use solutions to relaxations of subproblems, and branch-and-cut strategies for solving the integer program formulation of this problem [Chi01, Man94, Pfe08]. These methods are not row-action methods and generally require operating on the entire system or large subsystems, making them impractical for our setting. It has been previously observed that the behavior of projection and relaxation methods can detect inconsistent systems and thermal variants of these methods have been developed for identifying consistent subsystems and even for approximating MAX-FS solutions [JCC15, ABH05]. These methods are row-action methods, but are designed for MAX-FS problems rather than the large consistent subsystem setting we have described, and a comparison is not natural.

From this viewpoint, our problem is that of solving a consistent *subsystem* of equations where we assume the size of this system is large relative to the size of the entire inconsistent system. We are motivated by the setting in which one must solve an overdetermined system of equations in which few of the rows have been corrupted. Often, in applications, one is not concerned with finding the maximal feasible subsystem, but instead finding its solution, and this subsystem can be assumed

to be large. Applications in this framework include logic programming, error detection and correction in telecommunications, and infeasible linear programming models. Our approach to solving for \mathbf{x}^* will make use of the randomized Kaczmarz method, which we discuss next.

The Kaczmarz method is a popular iterative solver for overdetermined systems of linear equations and is especially preferred for systems with an extremely large number of rows. The method consists of sequential orthogonal projections toward the solution set of a single equation (or subsystem). Given the system $A\mathbf{x} = \mathbf{b}$, the method computes iterates by projecting onto the hyperplane defined by the equation $\mathbf{a}_i^T \mathbf{x} = b_i$, where \mathbf{a}_i^T is a selected row of the matrix A and b_i is the corresponding entry of \mathbf{b} . The iterates are recursively defined as

$$(1) \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \frac{b_i - \mathbf{a}_i^T \mathbf{x}_k}{\|\mathbf{a}_i\|^2} \mathbf{a}_i,$$

where \mathbf{a}_i is selected from among the rows of A . The seminal work [SV09] proved exponential convergence for the randomized Kaczmarz method where the rows \mathbf{a}_i are chosen with probability $\|\mathbf{a}_i\|^2 / \|A\|_F^2$. Since then many variants of the method have been proposed and analyzed for various types of systems; see, e.g., [GR15, NT14, ZF13, EN11, LL10].

It is known that the randomized Kaczmarz method converges for systems $A\mathbf{x} = \mathbf{b}$ corrupted by noise with an error threshold dependent on A and the noise. In [Nee10] it was shown that this method has iterates that satisfy

$$(2) \quad \|\mathbf{x}_k - \mathbf{x}_{\text{LS}}\|^2 \leq \left(1 - \frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right)^k \|\mathbf{x}_0 - \mathbf{x}_{\text{LS}}\|^2 + \frac{\|A\|_F^2}{\sigma_{\min}^2(A)} \|\mathbf{e}\|_\infty^2,$$

where $\sigma_{\min}(A)$ denotes the minimum singular value of A , $\|A\|_F$ its Frobenius norm, \mathbf{x}_{LS} the least squares solution and $\mathbf{e} = \mathbf{b} - A\mathbf{x}_{\text{LS}}$ denotes the error term (also known as the residual). There are variants of this method that converge to the least squares solution [CEG83, ZF13]; however, these typically either require operations on the columns or unknown relaxation parameters. Additionally, it is known that if a linear system of equations or inequalities is feasible, then randomized Kaczmarz will provide a *proof* or *certificate of feasibility*, and there are probabilistic guarantees on how quickly it will do so [DLHN17]. However, we are now interested in using randomized Kaczmarz for infeasible systems in which the least-squares solution is unsatisfactory because it is far from satisfying most of the equations (e.g., the noise is sparse and large).

1.1. Contribution. We develop methods that seek to identify the corrupted entries in \mathbf{b} and then converge to the pseudosolution. Our methods consist of several “rounds” of many iterations of the Randomized Kaczmarz (RK) method. The intuition behind these methods is that if there are only few corrupted equations and many consistent equations, then the iterations of RK will select consistent equations with high probability, producing an iterate near the pseudosolution and then the largest residual entries will correspond to the corrupted equations. We give a lower bound on the probability that a single round detects the corrupted equations. One may run many independent rounds and increase the probability of detecting these corrupted constraints. We then give a lower bound on the probability that one of these many rounds will detect the corrupted equations.

1.2. Notation. To simplify, we define some general notation to be used throughout the paper. Let $\|\cdot\|$ refer to the Euclidean norm. We will denote vectors in boldface (e.g., \mathbf{x}), and matrices and scalars in nonbold (e.g., A and b_i). We use $\mathbf{a}_i^T \in \mathbb{R}^n$ to represent the i th row of $A \in \mathbb{R}^{m \times n}$ and $\mathbf{e}_i \in \mathbb{R}^m$ to represent the i th coordinate vector. We will denote the origin as $\mathbf{0} \in \mathbb{R}^n$. Let $[m] = \{1, 2, \dots, m\}$ and let $[A]$ refer to the set of indices of the rows of matrix A (i.e., for $A \in \mathbb{R}^{m \times n}$, $[A] = [m]$). For $D \subset [A]$, we let $A_{D^c} = A_{[A]-D}$ be the submatrix of A whose rows are indexed by the complement of D . Denote the minimum singular value of A as $\sigma_{\min}(A)$.

As mentioned above, we consider the situation in which $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$ define an inconsistent system of equations, but there is a large consistent subsystem. That is, A and $\mathbf{b}^* \in \mathbb{R}^m$ define the consistent system of equations with solution $\mathbf{x}^* \in \mathbb{R}^n$ (i.e., $A\mathbf{x}^* = \mathbf{b}^*$) which we will refer to as the pseudosolution of the system of equations defined by A and \mathbf{b} . The right-hand side vector \mathbf{b}^* is corrupted by \mathbf{b}_C , so $\mathbf{b} = \mathbf{b}^* + \mathbf{b}_C$. Let $I \subset [m]$ be the set of indices of inconsistent equations, i.e., $\text{supp}(\mathbf{b}_C) = I$ and $s := |I| \ll m$. We refer to the amount of corruption in each index of I by $\epsilon_i \in \mathbb{R}$, so $\mathbf{b}_C = \sum_{i \in I} \epsilon_i \mathbf{e}_i$. We let ϵ^* be the smallest absolute entry of the corruption, $\epsilon^* := \min_{i \in I} |\epsilon_i|$. We will also use A_* to refer to the matrix A without the rows indexed by I , $A_* = A_{I^c}$, and likewise for \mathbf{b}_* . Note then that $\mathbf{b}_* := \mathbf{b}_{I^c} = \mathbf{b}_{I^c}^*$. For convenience, we will assume throughout the paper that the rows of A are normalized to have unit norm.

2. A Kaczmarz-type approach for corrupted systems. We consider here solving a consistent system of linear equations that has been corrupted, $A\mathbf{x} = \mathbf{b}^* + \mathbf{b}_C$ with $\|\mathbf{b}_C\|_0 = s \ll m$. Formally, given matrix A and right-hand side vector \mathbf{b} , we are searching for \mathbf{x}^* given by

$$(3) \quad (\mathbf{b}_C, \mathbf{x}^*) = \operatorname{argmin}_{\mathbf{b}_C, \mathbf{x}} \|\mathbf{b}_C\|_0 \quad \text{such that} \quad A\mathbf{x} = \mathbf{b} - \mathbf{b}_C.$$

One can design pathological examples of corrupted linear systems in which the solution to (3) differs from the pseudosolution; however, typically these solutions coincide. In particular, for \mathbf{a}_i in general position, this holds. First, we recall the RK method and fundamental convergence results.

Method 1. Randomized Kaczmarz [SV09].

```

1: procedure RK( $A, \mathbf{b}, \mathbf{x}_0, k$ )
2:   for  $j = 1, 2, \dots, k$  do
3:      $\mathbf{x}_j = \mathbf{x}_{j-1} + \frac{b_{i_j} - \mathbf{a}_{i_j}^T \mathbf{x}_{j-1}}{\|\mathbf{a}_{i_j}\|^2} \mathbf{a}_{i_j}$ , where  $i_j = t \in [m]$  with probability proportional to  $\|\mathbf{a}_t\|^2$ 
4:   end for
5:   return  $\mathbf{x}_k$ 
6: end procedure
```

THEOREM 1 (see [SV09]). *Let \mathbf{x} be the solution of $A\mathbf{x} = \mathbf{b}$, then RK converges to \mathbf{x} in expectation with the average error*

$$\mathbb{E}\|\mathbf{x}_k - \mathbf{x}\|^2 \leq \left(1 - \frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right)^k \|\mathbf{x}_0 - \mathbf{x}\|^2.$$

The intuition behind our proposed approach is simple. Since the number of corruptions is small, most iterates of an RK approach will be close to the pseudosolution,

since it is rare to project onto a corrupted hyperplane. Therefore, if we run the RK method several times, or for several *rounds* of iterations, most of the iterates upon which we halt will be close to the pseudosolution. Such iterates will also have the property that the largest components of their residual, $|A\mathbf{x}_k - \mathbf{b}|$, will correspond to the large corrupted entries. We can thus utilize this knowledge to gradually detect the corruptions, remove them from the system, and solve for the desired pseudosolution.

Our proposed methods can thus be described as follows. Each method consists of W rounds of k RK iterations beginning with $\mathbf{x}_0 = \mathbf{0}$. In each round, we collect the d indices of the largest magnitude residual entries and after all rounds, we solve the system without the rows of A indexed by these collected indices (there may be as many as dW rows removed). The methods differ in two ways. First, we can choose to remove d rows within each round (resulting in Method 2 below), or simply collect these indices and remove all collected rows after the W rounds (resulting in Methods 3 and 4 below). Second, when waiting to remove the rows until after W rounds, we may simply select the d largest residual entries in each round (Method 3), or we may require that the selected indices are always unique (so exactly dW rows are removed), resulting in Method 4. The values W, k , and d are all parameters of the methods. We give theoretical results for various natural choices of these parameters.

Method 2. Multiple Round Kaczmarz with Removal.

```

1: procedure MRKwR( $A, \mathbf{b}, k, W, d$ )
2:    $B = A, \mathbf{c} = \mathbf{b}$ 
3:   for  $i = 1, 2, \dots, W$  do
4:      $\mathbf{x}_k^i = RK(B, \mathbf{c}, \mathbf{0}, k)$ 
5:      $D = \operatorname{argmax}_{D \subset [B], |D|=d} \sum_{j \in D} |B\mathbf{x}_k^i - \mathbf{c}|_j$ 
6:      $B = B_{D^c}, \mathbf{c} = \mathbf{c}_{D^c}$ 
7:   end for
8:   return  $\mathbf{x}$ , where  $B\mathbf{x} = \mathbf{c}$ 
9: end procedure

```

Method 3. Multiple Round Kaczmarz without Removal.

```

1: procedure MRKwOR( $A, \mathbf{b}, k, W, d$ )
2:    $S = \emptyset$ 
3:   for  $i = 1, 2, \dots, W$  do
4:      $\mathbf{x}_k^i = RK(A, \mathbf{b}, \mathbf{0}, k)$ 
5:      $D = \operatorname{argmax}_{D \subset [A], |D|=d} \sum_{j \in D} |A\mathbf{x}_k^i - \mathbf{b}|_j$ 
6:      $S = S \cup D$ 
7:   end for
8:   return  $\mathbf{x}$ , where  $A_{S^c}\mathbf{x} = \mathbf{b}_{S^c}$ 
9: end procedure

```

2.1. Main results. Our theoretical results provide a lower bound for the probability of successfully removing all corrupted equations after performing Method 3 or Method 4 with natural values for k, d , and W . Lemma 1 shows that there is a *detection horizon* around the pseudosolution, so that if $\|\mathbf{x} - \mathbf{x}^*\|$ is sufficiently small, the largest residual entries (of $|A\mathbf{x} - \mathbf{b}|$) correspond exactly to the corrupted equations and we may distinguish these equations from the consistent system. Lemma 2 gives a value of k so that after k iterations of RK, one can give a nonzero lower bound on

Method 4. Multiple Round Kaczmarz without Removal with Unique Selection.

```

1: procedure MRKwoRUS( $A, \mathbf{b}, k, W, d$ )
2:    $S = \emptyset$ 
3:   for  $i = 1, 2, \dots, W$  do
4:      $\mathbf{x}_k^i = RK(A, \mathbf{b}, \mathbf{0}, k)$ 
5:      $D = \operatorname{argmax}_{D \subset [A] - S, |D|=d} \sum_{j \in D} |A\mathbf{x}_k^i - \mathbf{b}|_j$ 
6:      $S = S \cup D$ 
7:   end for
8:   return  $\mathbf{x}$ , where  $A_{S^c} \mathbf{x} = \mathbf{b}_{S^c}$ 
9: end procedure

```

the probability that the current iterate is within the detection horizon. Theorems 2 and 3 then give lower bounds on the probability of successfully detecting all corrupted equations in one out of all W rounds for Methods 3 and 4, respectively. Proofs of all results are contained in the appendix.

LEMMA 1. *If $\|\mathbf{x} - \mathbf{x}^*\| < \frac{1}{2}\epsilon^*$, we have that the $d \leq s$ indices of largest magnitude residual entries are contained in I ; that is, for*

$$D = \operatorname{argmax}_{D \subset [A], |D|=d} \sum_{i \in D} |A\mathbf{x} - \mathbf{b}|_i$$

we have $D \subset I$.

LEMMA 2. *Let $0 < \delta < 1$. Define*

$$k^* = \max \left(0, \left\lceil \frac{\log \left(\frac{\delta(\epsilon^*)^2}{4\|\mathbf{x}^*\|^2} \right)}{\log \left(1 - \frac{\sigma_{\min}^2(A_*)}{m-s} \right)} \right\rceil \right).$$

Then in round 1 of Method 3 or Method 4, the iterate produced by the RK iterations, $\mathbf{x}_{k^}^i$ satisfies*

$$(4) \quad \mathbb{P} \left[\|\mathbf{x}_{k^*}^i - \mathbf{x}^*\| \leq \frac{1}{2}\epsilon^* \right] \geq (1 - \delta) \left(\frac{m-s}{m} \right)^{k^*}.$$

First, note that we must restrict k^* to be nonnegative; since $\log \left(1 - \frac{\sigma_{\min}^2(A_*)}{m-s} \right)$ is negative, if $\log \left(\frac{\delta(\epsilon^*)^2}{4\|\mathbf{x}^*\|^2} \right)$ is positive, we must define $k^* = 0$. However, this corresponds to the situation in which $\epsilon^* > 2\|\mathbf{x}^*\|$ and the initial iterate $\mathbf{x}_0 = \mathbf{0}$ is within the detection horizon. Additionally, note that k^* depends upon δ , so one is not able to make this probability as large as one likes. As δ decreases, k^* increases, so the right-hand side of (4) is bounded away from 1. In Figure 2, we plot k^* and $(1 - \delta) \left(\frac{m-s}{m} \right)^{k^*}$ for Gaussian systems with various number of corruptions. In the plots, we see that the value of δ which maximizes this probability depends upon s . Determining this maximizing δ was not computable in closed form. Additionally, we point out that the empirical behavior of the method does not appear to depend upon δ ; we believe this is an artifact of our proof.

THEOREM 2. *Let $0 < \delta < 1$. Suppose $d \geq s$, $W \leq \lfloor \frac{m-n}{d} \rfloor$, and k^* is as given in Lemma 2. Then Method 3 on A, \mathbf{b} will detect the corrupted equations ($I \subset S$) and the*

remaining equations given by $A_{[m]-S}, \mathbf{b}_{[m]-S}$ will have solution \mathbf{x}^* with probability at least

$$1 - \left[1 - (1 - \delta) \left(\frac{m-s}{m} \right)^{k^*} \right]^W.$$

In Figure 2, we plot $1 - \left[1 - (1 - \delta) \left(\frac{m-s}{m} \right)^{k^*} \right]^W$ for corrupted Gaussian systems and choices of δ . Here $W = \lfloor (m-n)/d \rfloor$ and $d = s$. Again, we reiterate that we believe the dependence upon δ is an artifact of the proof of Lemma 2. Substituting e.g., $\delta = 0.5$ in probability bounds gives a value not far from its maximum for all systems we studied; see Figures 2 and 3.

THEOREM 3. *Let $0 < \delta < 1$. Suppose $d \geq 1$, $W \leq \lfloor \frac{m-n}{d} \rfloor$, and k^* is as given in Lemma 2. Then Method 4 on A, \mathbf{b} will detect the corrupted equations ($I \subset S$) and the remaining equations given by $A_{[m]-S}, \mathbf{b}_{[m]-S}$ will have solution \mathbf{x}^* with probability at least*

$$1 - \sum_{j=0}^{\lceil s/d \rceil - 1} \binom{W}{j} p^j (1-p)^{W-j},$$

where $p = (1 - \delta) \left(\frac{m-s}{m} \right)^{k^*}$.

We are not able to prove a result similar to Theorem 2 or Theorem 3 for Method 2 due to the fact that rounds of this method are not independent because one removes equations after each round.

In Figure 4, we plot $1 - \sum_{j=0}^{\lceil s/d \rceil - 1} \binom{W}{j} p^j (1-p)^{W-j}$ for corrupted Gaussian systems and choices of δ . Here $W = 2$, $d = \lceil s/2 \rceil$, and k^* is as given in Lemma 2. We believe that the dependence upon δ is an artifact of our proof. Evidence suggesting this is seen in the middle and right plots of Figure 4, as the empirical behavior of Method 4 does not appear to depend upon δ .

These bounds on the probability of successfully detecting all corrupted equations in one round, while provable and nonzero, are pessimistic and do not resemble the experimental rate of success for any systems we studied; see Figures 2 and 3. A tighter bound on the rate of convergence for particular systems could provide a tighter lower bound on this probability.

3. Experimental results. We are only able to prove theoretical results when the rounds of Methods 3 and 4 are independent and for the specified values of k^* and d . However, in practice, these methods perform well for different values of k and d , and Method 2 can be quite successful. In this section, we present experimental results demonstrating the performance of these methods, for various choices of d and k , on Gaussian, correlated, and real systems.

We plot our theoretical bounds as well as comparable empirical measures, which we denote “success rates.” Note that Theorem 2 provides a bound on the probability that in *one* of the rounds of Methods 3 we successfully detect *all* of the corrupted equations. For this reason, in Figures 2 and 3, we plot the empirical rate at which our method selects all of the s corrupted equations in one of the W rounds over 100 trials. However, in Figures 5 and 6, we plot the rates at which all of the s corrupted equations are selected over *all* of the W rounds over 100 trials, which is a more practical measure of success. However, Theorem 3 presents a bound on the probability that all of the s corrupted equations are selected after *all* of the W rounds in Method 4. Figure 4 plots this bound alongside the corresponding empirical rate. The measure of success plotted in each figure is defined in the figure caption.

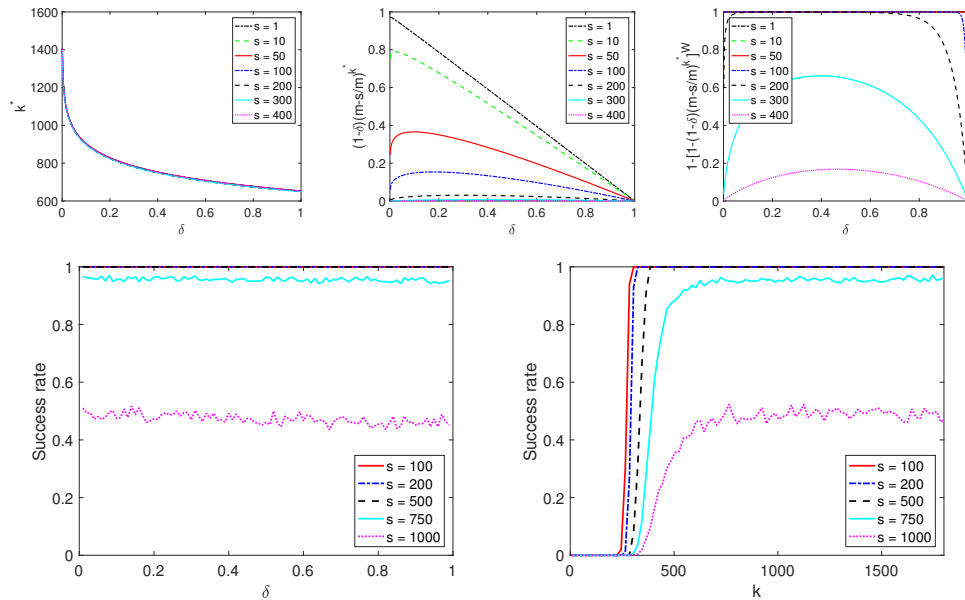


FIG. 2. Plots for Method 3 on 50000×100 Gaussian system (normalized) with various numbers of corrupted equations, s . Upper left: k^* as given in Lemma 2. Upper middle: lower bound on probability of successfully detecting all corrupted equations in single round as given in Lemma 2. Upper right: Lower bound on probability of successfully detecting all corrupted equations in one round out of $W = \lfloor \frac{m-n}{s} \rfloor$ rounds as given in Theorem 2. Lower left: Experimental ratio of success of detecting all s corrupted equations in one round out of $W = \lfloor \frac{m-n}{s} \rfloor$ for choice of δ . Lower right: Experimental ratio of success of detecting all s corrupted equations in one round out of $W = \lfloor \frac{m-n}{s} \rfloor$ for choice of k (number of RK iterations per round).

3.1. Random data experiments. The plots in Figures 2 and 5 are all for Method 3 on a 50000×100 Gaussian system defined by A with $a_{ij} \sim \mathcal{N}(0, 1)$, then normalized. The system is corrupted in randomly selected right-hand side entries with random integers in $[1, 5]$ so that $\epsilon^* = 1$. For these plots and experiments, $d = s$. The upper left image of Figure 2 plots the k^* values defined in Lemma 2 for this system, and the upper middle image plots the theoretically guaranteed probability of selecting all s corrupted equations in a single round. The upper right image of Figure 2 plots the theoretically guaranteed probability of selecting all s corrupted equations in one round out of the $W = \lfloor \frac{m-n}{s} \rfloor$, while the lower left image plots the ratio of successful trials, in which all s corrupted equations were selected in one round of the W , out of 100 trials. The lower right plot of Figure 2 plots how this ratio changes as the number of RK iterations, k , in each round varies. Finally, Figure 5 plots the ratio of successful trials, in which all s corrupted equations were selected after all $W = \lfloor \frac{m-n}{s} \rfloor$ rounds, out of 100 trials as one varies δ (left) and k (right). We note that the lower bounds on the probability of successfully detecting all corrupted equations in one round are quite pessimistic; experimentally (in the lower left plot) we see that Method 3 is able to detect all corruption for much larger numbers of corrupted equations, s , than predicted theoretically (in the upper right plot). Additionally, we note that experimentally, successfully detecting the corrupted equations does not appear to depend upon δ . For all $0 < \delta < 1$, the k^* value defined in Lemma 2 appears to be large enough to guarantee convergence within the detection horizon.

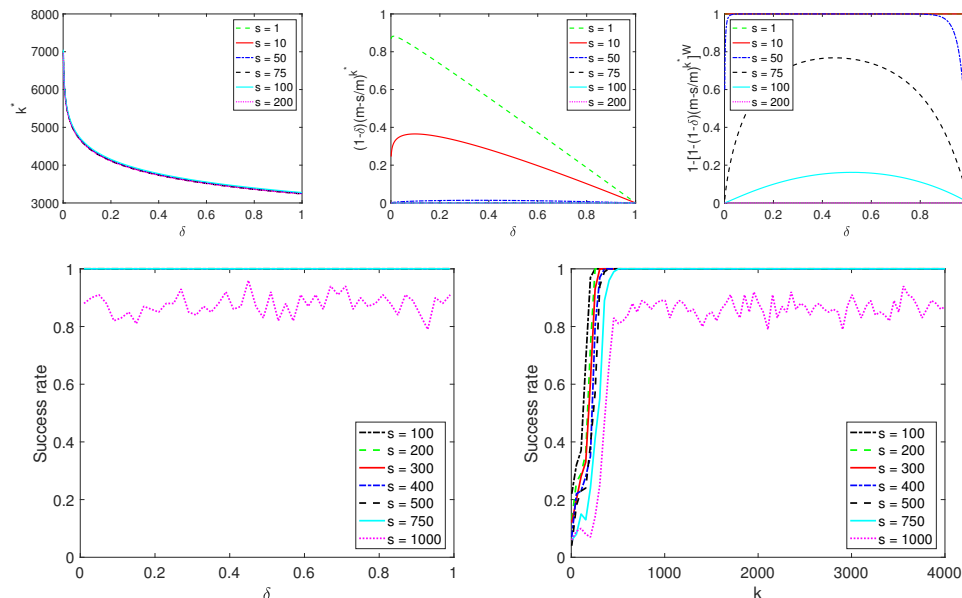


FIG. 3. Plots for Method 3 on 50000×100 correlated system (normalized) with various numbers of corrupted equations, s . Upper left: k^* as given in Lemma 2. Upper middle: Lower bound on probability of successfully detecting all corrupted equations in single round as given in Lemma 2. Upper right: Lower bound on probability of successfully detecting all corrupted equations in one round out of $W = \lfloor \frac{m-n}{s} \rfloor$ rounds as given in Theorem 2. Lower left: Experimental ratio of success of detecting all s corrupted equations in one round out of $W = \lfloor \frac{m-n}{s} \rfloor$ for choice of δ . Lower right: Experimental ratio of success of detecting all s corrupted equations in one round out of $W = \lfloor \frac{m-n}{s} \rfloor$ for choice of k (number of RK iterations per round).

In Figure 4, we briefly explore the theoretical guarantees for Method 4 given in Theorem 3, and the empirical behavior of this method. These plots are for a 50000×100 Gaussian system (normalized) with various number, s , of corrupted equations. We randomly sample s entries of the right-hand side vector b and corrupt them by adding 1, so $\epsilon^* = 1$. The plot on the left of Figure 4 plots the lower bound on the probability of selecting all corrupted equations given in Theorem 3 for $W = 2$, $d = \lceil s/2 \rceil$, and k^* as given in Lemma 2. Meanwhile, in the middle and right plots of Figure 4, we plot the average fraction of corrupted equations recorded for Method 4 with $W = \lceil s/d \rceil$ and $d = \lceil s/10 \rceil$ for 100 trials. The middle plot has k^* (as given in Lemma 2) RK iterations per round for varying δ , while the right plot has varying k (number of RK iterations per round). Note that this experiment is different from the others we present in this section as we display the average fraction of corrupted equations recorded over 100 trials, rather than the fraction of trials which detected all corrupted equations. We note that the theoretical bound plotted on the left of Figure 4 is even more pessimistic than of Method 3, but meanwhile the empirical performance of Method 4 plotted in the middle and right of Figure 4 is even better than that seen for Method 3. For this reason, we do not plot these bounds (Theorem 3) or the performance of Method 4 for additional systems as we expect the results to trend similarly for other systems.

The figures for Method 3 mentioned above are recreated for a system whose rows are more correlated ($A \in \mathbb{R}^{50000 \times 100}$ with $a_{ij} \sim \mathcal{N}(1, 0.5)$ then normalized) in Figures

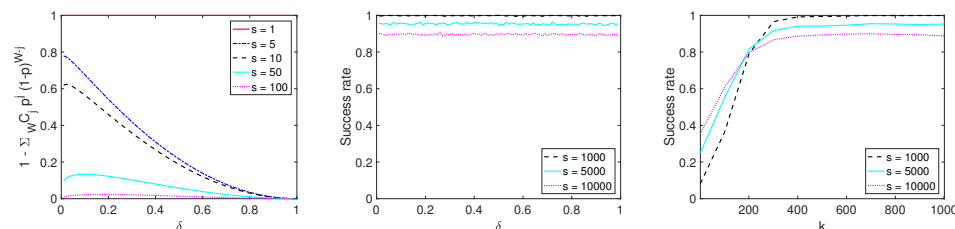


FIG. 4. Plots for Method 4 on 50000×100 Gaussian system (normalized) with various number of corrupted equations, s . Left: Bound given in Theorem 3 for Method 4 with $W = 2$, $d = \lceil s/2 \rceil$, and k^* as given in Lemma 2. Middle: Average fraction of corrupted equations detected after $W = \lceil s/d \rceil$ rounds recording $d = \lceil s/10 \rceil$ equations per round with k^* (as given in Lemma 2) RK iterations per round for varying δ in 100 trials. Right: Average fraction of corrupted equations detected after $W = \lceil s/d \rceil$ rounds recording $d = \lceil s/10 \rceil$ equations per round with varying k (number of RK iterations per round) in 100 trials.

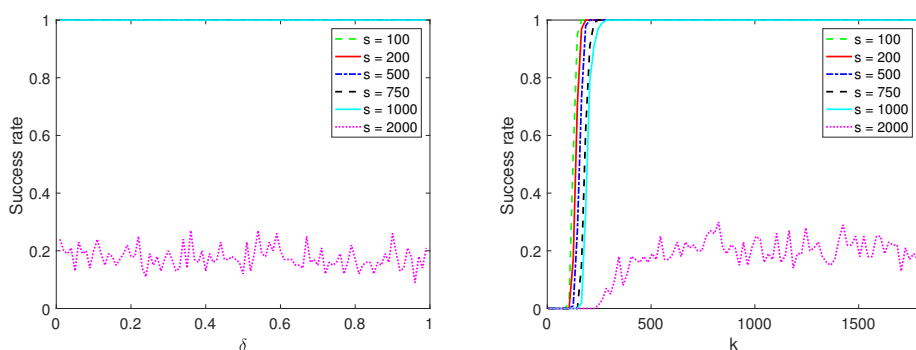


FIG. 5. Plots for Method 3 on 50000×100 Gaussian system (normalized) with various numbers of corrupted equations, s . Left: Experimental ratio of successfully detecting all s corrupted equations after all $W = \lfloor \frac{m-n}{s} \rfloor$ rounds for choice of δ . Right: Experimental ratio of successfully detecting all s corrupted equations after all $W = \lfloor \frac{m-n}{s} \rfloor$ rounds for choice of k (number of RK iterations per round).

3 and 6. The system is corrupted in randomly selected right-hand side entries with random integers in $[1, 5]$ so that $\epsilon^* = 1$. For these plots and experiments, $d = s$. The upper left image of Figure 3 plots the k^* values defined in Lemma 2 for this system, and the upper middle image plots the theoretically guaranteed probability of selecting all s corrupted equations in a single round of Method 3. The upper right image of Figure 3 plots the theoretically guaranteed probability of selecting all s corrupted equations in one round out of the $W = \lfloor \frac{m-n}{s} \rfloor$, while the lower left image plots the ratio of successful trials, in which all s corrupted equations were selected in one round of the W , out of 100 trials. The lower right plot of Figure 3 plots how this ratio changes as the number of RK iterations, k , in each round varies. Finally, Figure 6 plots the ratio of successful trials, in which all s corrupted equations were selected after all W rounds, out of 100 trials as one varies δ (left) and k (right). We note that the discrepancy between the lower bound on the probability of successfully detecting all corrupted equations in one round (upper right plot) and the experimental rate of detecting all corruption (lower left plot) is even larger than in the case of Gaussian systems.

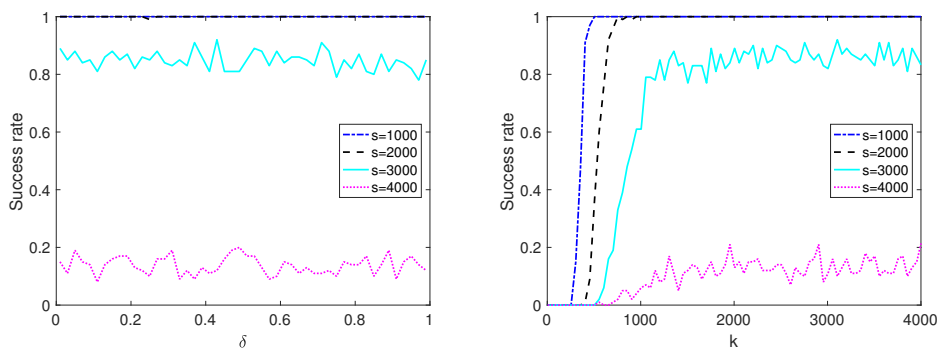


FIG. 6. Plots for Method 3 on 50000×100 correlated system (normalized) with various numbers of corrupted equations, s . Left: Experimental ratio of successfully detecting all s corrupted equations after all $W = \lfloor \frac{m-n}{s} \rfloor$ rounds for choice of δ . Right: Experimental ratio of successfully detecting all s corrupted equations after all $W = \lfloor \frac{m-n}{s} \rfloor$ rounds for choice of k (number of RK iterations per round).

First, note that k^* , as given in Lemma 2, depends very weakly upon s . In the upper left plots of Figures 2 and 3, the values of k^* plotted are very slightly different for different values of s (the line thickness makes these distinct lines appear as one). Note that the definition of k^* (the theoretically required number of RK iterations to reach the detection horizon) is defined by the theoretical convergence rate which can be quite pessimistic. As has been seen in the lower right plots of Figures 2 and 3, and in the right plots of Figures 5 and 6, detection can be successful with a significantly smaller choice of k . Note that in Figure 2, the theoretically required k^* value is between 600 and 1400 but $k > 500$ seems to perform well. Likewise, in Figure 3, the theoretically required k^* value is between 3000 and 8000 but $k > 500$ seems to perform well. It is unsurprising that this bound is even more pessimistic for the correlated system, as the conditioning of a correlated system causes the RK convergence guarantee to be quite poor, while experimentally we see a much faster rate of convergence.

3.2. Implementation considerations. There are several options for d , some more practically feasible than others. Our theoretical results are probabilistic guarantees for Method 3 with $d \geq s$, which, of course, cannot be known in practice, as well as for Method 4 with $d \geq 1$, which is practical, but the method is more expensive computationally. In practice, one could choose d as the user estimate for s .

The choice of d and W are complementary in that increasing d decreases W (since one may have less rounds if in each round more equations are selected). In selecting d and W , we wish to balance the desire to increase d in order to record all of the corrupted equations when we have a successful round with the fact that as d grows, we can have less rounds. We never discard or record more than $m - n$ of the constraints, as at the end of any method, we wish to have a full rank linear system of equations remaining whose solution is \mathbf{x}^* , the pseudosolution. Thus, for any d we may not run more than $W = \lfloor \frac{m-n}{d} \rfloor$ rounds. However, in practice, this choice of W may be larger than is necessary. This is explored in Figures 7 and 8. In the experiment producing Figure 7, we ran $W = \lfloor \frac{m-n}{d} \rfloor$ rounds of Method 3 with k^* (defined in Lemma 2) RK iterations selecting d equations each round, and record the ratio of successful trials, which selected all s corrupted equations after all rounds, out of 100

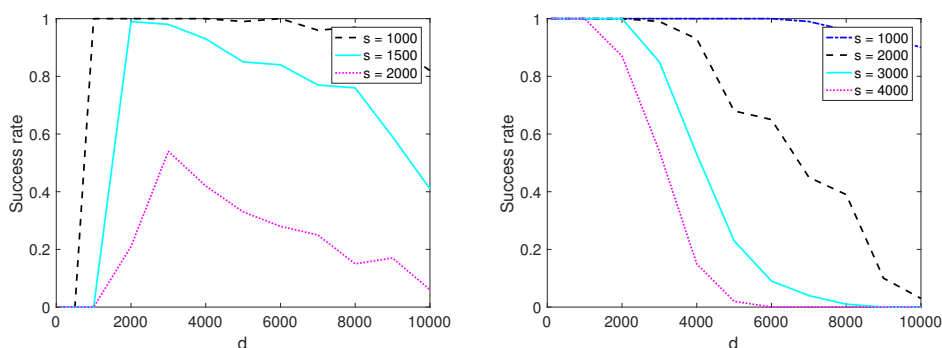


FIG. 7. *Left: Experimental ratio of success of detecting all s corrupted equations after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds of k^* (as given in Lemma 2) RK iterations selecting d equations for 50000×100 Gaussian system with s corrupted equations. Right: Experimental ratio of success of detecting all s corrupted equations after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds of k^* (as given in Lemma 2) RK iterations selecting d equations for 50000×100 correlated system with s corrupted equations.*

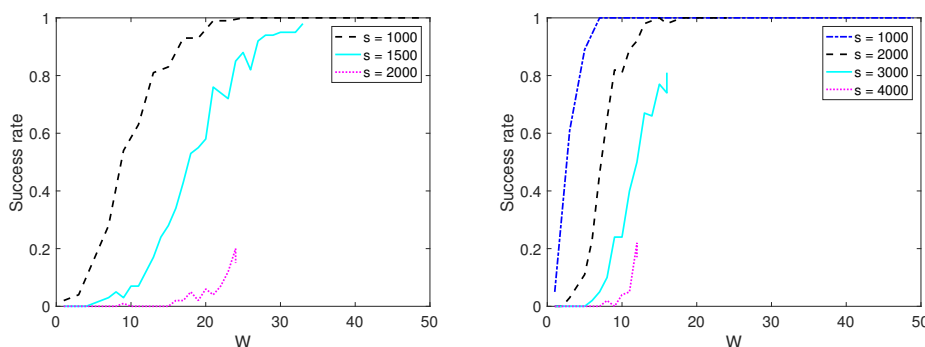


FIG. 8. *Left: Experimental ratio of success of detecting all s corrupted equations after all $W \leq \lfloor \frac{m-n}{s} \rfloor$ rounds of k^* (as given in Lemma 2) RK iterations for 50000×100 Gaussian system with s corrupted equations. Right: Experimental ratio of success of detecting all s corrupted equations after all $W \leq \lfloor \frac{m-n}{s} \rfloor$ rounds of k^* (as given in Lemma 2) RK iterations for 50000×100 correlated system with s corrupted equations.*

trials. The figure on the left shows the results for a Gaussian system, while the figure on the right shows the results for a correlated system. In the experiment producing Figure 8, we ran $W \leq \lfloor \frac{m-n}{s} \rfloor$ rounds of Method 3 with k^* (defined in Lemma 2) RK iterations selecting s equations each round, and recorded the ratio of successful trials, which selected all s corrupted equations after all rounds, out of 100 trials. The figure on the left is for a Gaussian system, while the figure on the right is for a correlated system. Both are corrupted with random integers in $[1, 5]$ in randomly selected entries of \mathbf{b} , so $\epsilon^* = 1$.

Method 2, despite not having independent rounds, performs well in practice as is seen in Figure 9. In this experiment, we perform $W = \lfloor \frac{m-n}{s} \rfloor$ rounds of Method 2 with k RK iterations, removing s equations each round. The plot shows the ratio of successful trials, in which all s corrupted equations are removed after all W rounds, out of 100 trials. The method is run on a Gaussian system which is corrupted by random integers in $[1, 5]$ in randomly selected entries of \mathbf{b} , so $\epsilon^* = 1$.

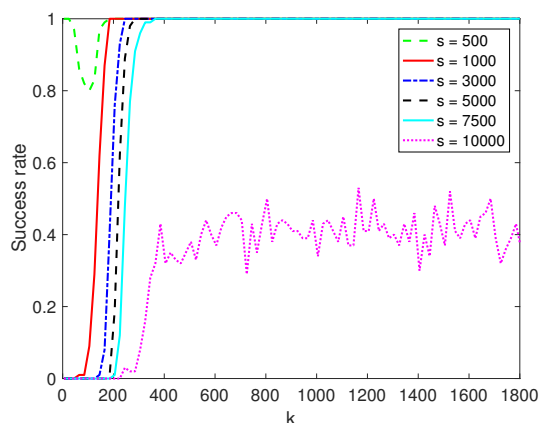


FIG. 9. Experimental ratio of success of removing (Method 2) all s corrupted equations after all $W = \lfloor \frac{m-n}{s} \rfloor$ rounds for 50000×100 Gaussian system with s corrupted equations and choice of k .

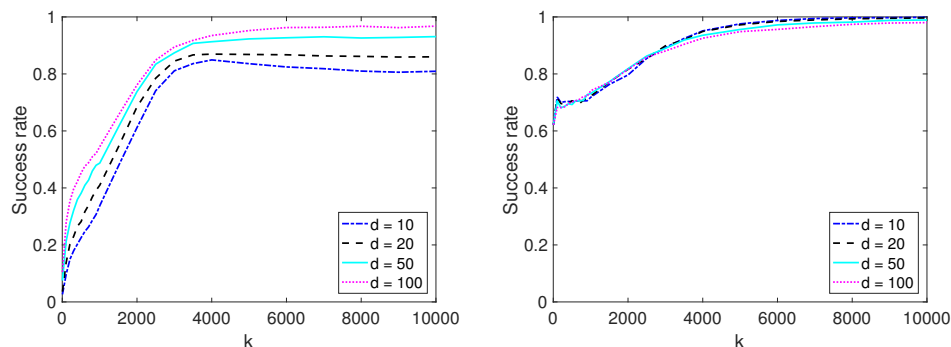


FIG. 10. Plots for 1200×400 tomography system with $s = 100$ corrupted equations. Left: average fraction of corrupted equations detected in 100 trials after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds of Method 3. Right: average fraction of corrupted equations removed in 100 trials after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds of Method 2.

3.3. Real data experiments. We additionally test the methods on real data. Our first experiments are on tomography problems, generated using the MATLAB Regularization Toolbox by P.C. Hansen (<http://www.imm.dtu.dk/~pcha/Regutools/>) [Han07]. In particular we present a 2D tomography problem $A\mathbf{x} = \mathbf{b}$ for an $m \times n$ matrix with $m = fN^2$ and $n = N^2$. Here A corresponds to the absorption along a random line through an $N \times N$ grid. In our experiments we set $N = 20$ and the over-sampling factor $f = 3$. This yielded a matrix A with condition number $\kappa(A) = 2.08$. As the resulting system was consistent, we randomly sampled $s = 100$ constraints uniformly from among the rows of A and corrupted the right-hand side vector \mathbf{b} by adding 1 in these entries, so $\epsilon^* = 1$. This corrupted system has $k^* = 66334$ (as given in Lemma 2). Figure 10 contains the average fraction of the $s = 100$ corrupted equations detected or removed for Methods 3 (left) and 2 (right) after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds for various values of k (RK iterations per round) for 100 trials.

We also generated corrupted data sets using the Wisconsin (Diagnostic) Breast Cancer data set, which includes data points whose features are computed from a

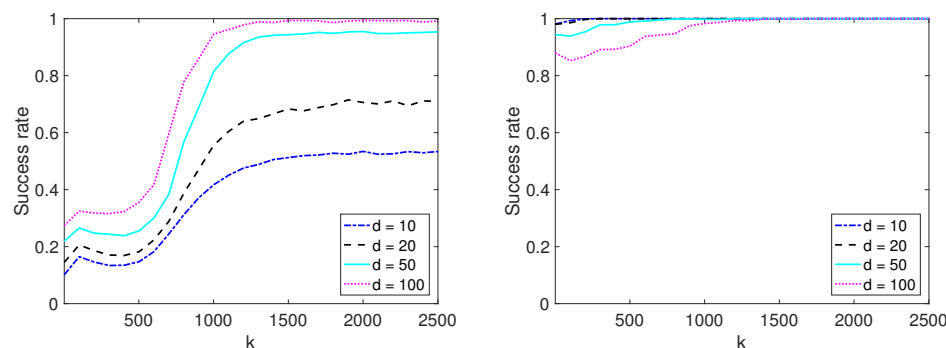


FIG. 11. Plots for 699×10 system defined by Wisconsin (Diagnostic) Breast Cancer data set with $s = 100$ corrupted equations. Left: average fraction of corrupted equations detected in 100 trials after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds of Method 3. Right: average fraction of corrupted equations removed in 100 trials after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds of Method 2.

digitized image of a fine needle aspirate (FNA) of a breast mass and describe characteristics of the cell nuclei present in the image [Lic13]. This collection of data points forms our matrix $A \in \mathbb{R}^{699 \times 10}$, we construct \mathbf{b} to form a consistent system, and then corrupt a random selection of 100 entries of the right-hand side by adding 1, so $\epsilon^* = 1$. This corrupted system has $k^* = 3432$ (as given in Lemma 2). Figure 11 contains the average fraction of the $s = 100$ corrupted equations detected or removed for Methods 3 (left) and 2 (right) after all $W = \lfloor \frac{m-n}{d} \rfloor$ rounds for various values of k (RK iterations per round) for 100 trials.

3.4. Comparison to existing methods. As previously mentioned, although related, a comparison of Methods 2, 3, and 4 to methods designed for MAX-FEAS are not natural. Methods for MAX-FEAS are designed for a much more general and harder class of problems than our proposed methods. Methods for MAX-FEAS seek to carefully identify the largest feasible subproblem, while our methods seek to identify and discard potentially corrupted equations.

However, one may consider our methods as iteratively computing a solution to the current system of equations with a sparse residual, and deleting those entries corresponding to the nonzero entries in the residual. Thus, we compare the behavior of our method to convex optimization methods on the problem reformulation $\min \|r\|_1$ s.t. $A\mathbf{x} - \mathbf{b} = \mathbf{r}$, as this reformulation should encourage sparsity in the residual. In these experiments, we run several rounds of k iterations of various convex optimization methods (initialized with the last iterate from the previous round) implemented in the built-in MATLAB `fmincon` function, and remove d equations corresponding to the largest entries of the computed residual, \mathbf{r} . We measure the CPU time (using MATLAB `cputime`) required to remove all corrupted equations, and compare this to the CPU time required by Method 2 with the same d and k values to remove all corrupted equations.

We test on the same problem data as described in subsection 3.3. In Table 1, we report the CPU time(s) required to remove all 100 corrupted equations by Method 2 with $k = 8000$ and $d = 10$, and the method described above for removing $d = 10$ equations with the algorithms “interior-point,” “active-set,” and “sqp” with $k = 10$ iterations.

TABLE 1

CPU time(s) required to remove all 100 corrupted equations by Method 2 with $k = 8000$ and $d = 10$, and the method described above for removing $d = 10$ equations with the algorithms “interior-point,” “active-set,” and “sqp” with $k = 10$ iterations.

	Method 2	interior-point	active-set	sqp
tomography	1.72	795.66	6435.21	3513.30
breast cancer	1.74	202.09	2643.15	507.98

4. Conclusion. We have presented a framework of methods for using randomized projection methods to detect and remove corruptions in a system of linear equations. We provide theoretical bounds on the probability that these methods will successfully detect and remove all corrupted equations. Moreover, we provide ample experimental evidence that these methods successfully detect corrupted equations and these results far surpass the theoretical guarantees.

Appendix A. Proofs of main results. We separate our main theoretical results from their proofs so as to minimize distraction from the progression of the results and plots of the probability bounds demonstrated.

Proof of Lemma 1. Suppose $\|\mathbf{x} - \mathbf{x}^*\| < \frac{1}{2}\epsilon^*$. Note that for $\|\mathbf{a}_i\| = 1$, we have

$$|r_i| = |\mathbf{A}\mathbf{x} - \mathbf{b}|_i = |\mathbf{a}_i^T \mathbf{x} - b_i| = \frac{|\mathbf{a}_i^T \mathbf{x} - b_i|}{\|\mathbf{a}_i\|} = d(\mathbf{x}, H_i),$$

where $d(\mathbf{x}, H)$ is the Euclidean distance of \mathbf{x} to the set H , and $H_i = \{\mathbf{x} : \mathbf{a}_i^T \mathbf{x} = b_i\}$ is the hyperplane defined by the i th equation. Next, note that

$$d(\mathbf{x}^*, H_i) = |\mathbf{a}_i^T \mathbf{x}^* - b_i| = |b_i^* - b_i| = \begin{cases} |\epsilon_i|, & i \in I, \\ 0, & i \notin I. \end{cases}$$

Now, consider r_i for $i \in I$. Denoting by P_H the orthogonal projection onto H , note that

$$\begin{aligned} |r_i| &= d(\mathbf{x}, H_i) = \|P_{H_i}(\mathbf{x}) - \mathbf{x}\| \\ &= \|P_{H_i}(\mathbf{x}) - \mathbf{x}^* - (\mathbf{x} - \mathbf{x}^*)\| \\ &\geq \| \|P_{H_i}(\mathbf{x}) - \mathbf{x}^*\| - \|\mathbf{x} - \mathbf{x}^*\| \| \\ &\geq d(\mathbf{x}^*, H_i) - \|\mathbf{x} - \mathbf{x}^*\| \\ &> \frac{1}{2}\epsilon^*, \end{aligned}$$

where the first inequality follows from the triangle inequality and the second from the fact that $\|P_{H_i}(\mathbf{x}) - \mathbf{x}^*\| \geq d(\mathbf{x}^*, H_i) = |\epsilon_i| \geq \epsilon^* > \|\mathbf{x} - \mathbf{x}^*\|$.

For $i \notin I$, since $\mathbf{x}^* \in H_i$,

$$r_i = d(\mathbf{x}, H_i) \leq \|\mathbf{x} - \mathbf{x}^*\| < \frac{1}{2}\epsilon^*.$$

To summarize,

$$|r_i| = |\mathbf{a}_i^T \mathbf{x}_k - b_i| \begin{cases} < \frac{1}{2}\epsilon^* & \text{for } i \notin I, \\ > \frac{1}{2}\epsilon^* & \text{for } i \in I. \end{cases}$$

Thus, if we consider the above,

$$D = \operatorname{argmax}_{D \subset [A], |D|=d} \sum_{i \in D} |A\mathbf{x} - \mathbf{b}|_i$$

is clearly a subset of I for $d \leq s$. \square

Proof of Lemma 2. Let E be the event that $i_1, i_2, \dots, i_{k^*} \notin I$ for all index selections in round W . Note that

$$\mathbb{P}(E) \geq \left(\frac{m-s}{m} \right)^{k^*}$$

since there are $m-s$ consistent equations and the equations are being selected uniformly at random.

Now, note that if one conditions upon E and looks at the expected value of $\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2$, this will be the same value as the expectation of $\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2$ if \mathbf{x}_{k^*} is created with RK run on A_*, \mathbf{b}_* ; we denote this expectation as $\mathbb{E}_{A_*, \mathbf{b}_*}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2]$. Applying Theorem 1, we see that

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2 | E] &= \mathbb{E}_{A_*, \mathbf{b}_*}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2] \\ &\leq \left(1 - \frac{\sigma_{\min}^2(A_*)}{m-s} \right)^{k^*} \|\mathbf{x}_0 - \mathbf{x}^*\|^2 \\ &= \left(1 - \frac{\sigma_{\min}^2(A_*)}{m-s} \right)^{k^*} \|\mathbf{x}^*\|^2. \end{aligned}$$

Now, since

$$k^* \geq \frac{\log \left(\frac{\delta(\epsilon^*)^2}{4\|\mathbf{x}^*\|^2} \right)}{\log \left(1 - \frac{\sigma_{\min}^2(A_*)}{m-s} \right)},$$

we have $\left(1 - \frac{\sigma_{\min}^2(A_*)}{m-s} \right)^{k^*} \leq \frac{\delta(\epsilon^*)^2}{4\|\mathbf{x}^*\|^2}$ and so

$$\mathbb{E}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2 | E] \leq \frac{\delta}{4}(\epsilon^*)^2.$$

Applying the conditional Markov inequality, we have

$$\begin{aligned} \mathbb{P}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2 > \tfrac{1}{4}(\epsilon^*)^2 | E] &\leq \frac{\mathbb{E}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2 | E]}{\tfrac{1}{4}(\epsilon^*)^2} \\ &\leq \frac{\tfrac{\delta}{4}(\epsilon^*)^2}{\tfrac{1}{4}(\epsilon^*)^2} = \delta. \end{aligned}$$

Thus, $\mathbb{P}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\|^2 \leq \tfrac{1}{4}(\epsilon^*)^2 | E] \geq 1 - \delta$ so

$$\mathbb{P}[\|\mathbf{x}_{k^*} - \mathbf{x}^*\| \leq \tfrac{1}{2}\epsilon^*] \geq (1 - \delta) \left(\frac{m-s}{m} \right)^{k^*}.$$

This completes the proof. \square

Proof of Theorem 2. Since $d \geq s$, we need only have one “successful” round where $\|\mathbf{x}_{k^*} - \mathbf{x}^*\| < \frac{1}{2}\epsilon^*$ in order to guarantee detection of all of the corrupted equations by Lemma 1. Since all of the rounds are independent from each other in Method 3 and by Lemma 2 the probability that $\|\mathbf{x}_{k^*} - \mathbf{x}^*\| < \frac{1}{2}\epsilon^*$ is at least $p := (1-\delta)\left(\frac{m-s}{m}\right)^{k^*}$, we may bound the probability of success by that of a binomial distribution with parameters W and p . Thus, success happens with probability at least

$$1 - \left[1 - (1-\delta)\left(\frac{m-s}{m}\right)^{k^*}\right]^W.$$

This completes the proof. \square

Proof of Theorem 3. Since $d \geq 1$ and we are selecting unique indices in each iteration of Method 4, we need to have $\lceil s/d \rceil$ “successful” rounds where $\|\mathbf{x}_{k^*} - \mathbf{x}^*\| < \frac{1}{2}\epsilon^*$ in order to guarantee detection of all of the corrupted equations by Lemma 1. Since all of the rounds of RK iterations are independent from each other in Method 4 and by Lemma 2 the probability that $\|\mathbf{x}_{k^*} - \mathbf{x}^*\| < \frac{1}{2}\epsilon^*$ is at least $p := (1-\delta)\left(\frac{m-s}{m}\right)^{k^*}$, we may bound the probability of success by that of a cumulative binomial distribution with parameters W and p and we calculate the probability that the number of successes, j , is at least $\lceil s/d \rceil$. Thus, success happens with probability defined by the probabilities that less than $\lceil s/d \rceil$ rounds are successful. The probability of success is bounded below by

$$1 - \sum_{j=0}^{\lceil s/d \rceil - 1} \binom{W}{j} p^j (1-p)^{W-j}.$$

This completes the proof. \square

REFERENCES

- [ABH05] E. AMALDI, P. BELOTTI, AND R. HAUSER, *Randomized relaxation methods for the maximum feasible subsystem problem*, in Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 3509, Springer, Berlin, 2005, pp. 249–264.
- [AK95] E. AMALDI AND V. KANN, *The complexity and approximability of finding maximum feasible subsystems of linear relations*, Theoret. Comput. Sci., 147 (1995), pp. 181–210.
- [CEG83] Y. CENSOR, P. P. B. EGGERMONT, AND D. GORDON, *Strong underrelaxation in Kaczmarz’s method for inconsistent systems*, Numer. Math., 41 (1983), pp. 83–92.
- [Chi01] J. W. CHINNECK, *Fast heuristics for the maximum feasible subsystem problem*, INFORMS J. Comput., 13 (2001), pp. 210–223.
- [DLHN17] J. A. DE LOERA, J. HADDOCK, AND D. NEEDELL, *A sampling Kaczmarz–Motzkin algorithm for linear feasibility*, SIAM J. Sci. Comput., 39 (2017), pp. S66–S87, <https://doi.org/10.1137/16M1073807>.
- [EK12] Y. C. ELДАР AND G. KUTYNIOK, *Compressed Sensing: Theory and Applications*, Cambridge University Press, Cambridge, UK, 2012.
- [EN11] Y. C. ELДАР AND D. NEEDELL, *Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma*, Numer. Algorithms, 58 (2011), pp. 163–177.
- [FR13] S. FOUCART AND H. RAUHUT, *A Mathematical Introduction to Compressive Sensing*, Birkhäuser/Springer, New York, 2013.
- [GHJ75] R. GORDON, G. T. HERMAN, AND S. A. JOHNSON, *Image reconstruction from projections*, Sci. Am., 233 (1975), pp. 56–61, 64–68.
- [GR15] R. M. GOWER AND P. RICHTÁRIK, *Randomized iterative methods for linear systems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1660–1690, <https://doi.org/10.1137/15M1025487>.
- [Han07] P. C. HANSEN, *Regularization tools version 4.0 for Matlab 7.3*, Numer. Algorithms, 46 (2007), pp. 189–194.

- [HLL78] G. T. HERMAN, A. LENT, AND P. H. LUTZ, *Relaxation methods for image reconstruction*, Comm. ACM, 21 (1978), pp. 152–158.
- [HN17] J. HADDOCK AND D. NEEDELL, *Randomized projections for corrupted linear systems*, AIP Conf. Proc., 1978 (2018), 470071.
- [JCC15] N. JAMIL, X. CHEN, AND A. CLONINGER, *Hildreth's algorithm with applications to soft constraints for user interface layout*, J. Comput. Appl. Math., 288 (2015), pp. 193–202.
- [Lic13] M. LICHMAN, *UCI Machine Learning Repository*, 2013, <http://archive.ics.uci.edu/ml>.
- [LL10] D. LEVENTHAL AND A. S. LEWIS, *Randomized methods for linear constraints: Convergence rates and conditioning*, Math. Oper. Res., 35 (2010), pp. 641–654.
- [Man94] O. L. MANGASARIAN, *Misclassification minimization*, J. Global Optim., 5 (1994), pp. 309–323.
- [MKC00] K. G. MURTY, S. N. KABADI, AND R. CHANDRASEKARAN, *Infeasibility analysis for linear systems, a survey*, Arab. J. Sci. Eng. Sect. C Theme Issues, 25 (2000), pp. 3–18.
- [Nat01] F. NATTERER, *The Mathematics of Computerized Tomography*, SIAM, Philadelphia, 2001, <https://doi.org/10.1137/1.9780898719284>.
- [Nee10] D. NEEDELL, *Randomized Kaczmarz solver for noisy linear systems*, BIT, 50 (2010), pp. 395–403.
- [NR08] Z. NUTOV AND D. REICHMAN, *Approximating maximum satisfiable subsystems of linear equations of bounded width*, Inform. Process. Lett., 106 (2008), pp. 203–207.
- [NT14] D. NEEDELL AND J. A. TROPP, *Paved with good intentions: Analysis of a randomized block Kaczmarz method*, Linear Algebra Appl., 441 (2014), pp. 199–221.
- [Pfe08] M. E. PFETSCH, *Branch-and-cut for the maximum feasible subsystem problem*, SIAM J. Optim., 19 (2008), pp. 21–38, <https://doi.org/10.1137/050645828>.
- [SV09] T. STROHMER AND R. VERSHYNIN, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl., 15 (2009), pp. 262–278.
- [ZF13] A. ZOUZIAS AND N. M. FRERIS, *Randomized extended Kaczmarz for solving least squares*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 773–793, <https://doi.org/10.1137/120889897>.

Preflight Results

Document Overview

Title: Randomized Projection Methods for Linear Systems with Arbitrary Covariate Sparsity | SIAM Journal on Scientific Computing
Author: Jamie Haddock and Deanna Needell
Creator: LaTeX with hyperref
Producer: pdfTeX-1.40.20; modified using iText 4.2.0 by 1T3XT

Preflight Information

Version: Qoppa jPDFPreflight v2020R1.03
Date: Sep 1, 2020 4:36:17 PM

Legend: (X) - Can NOT be fixed by PDF/A-1b conversion.
(!X) - Could be fixed by PDF/A-1b conversion. User chose to be warned in PDF/A settings.

Page 1 Results

(X) Font ArialMT is not embedded.

Page 2 Results

(X) Font ArialMT is not embedded.

Page 3 Results

(X) Font ArialMT is not embedded.

Page 4 Results

(X) Font ArialMT is not embedded.

Page 5 Results

(X) Font ArialMT is not embedded.

Page 6 Results

(X) Font ArialMT is not embedded.

Page 7 Results

(X) Font ArialMT is not embedded.

Page 8 Results

(X) Font ArialMT is not embedded.

Page 9 Results

(X) Font ArialMT is not embedded.

Page 10 Results

(X) Font ArialMT is not embedded.

Page 11 Results

(X) Font ArialMT is not embedded.

Page 12 Results

(X) Font ArialMT is not embedded.

Page 13 Results

(X) Font ArialMT is not embedded.

Page 14 Results

(X) Font ArialMT is not embedded.

Page 15 Results

(X) Font ArialMT is not embedded.

Page 16 Results

(X) Font ArialMT is not embedded.

Page 17 Results

(X) Font ArialMT is not embedded.

Page 18 Results

(X) Font ArialMT is not embedded.