ITERATIVE RANDOMIZED ROBUST LINEAR REGRESSION

Yannis Kopsinis[1] *Symeon Chouvardas*[2] *Sergios Theodoridis*[1]

[1]University of Athens,
Dept. of Informatics and Telecommunications,
Athens 15784, Greece.
Emails: kopsinis@ieee.org, stheodor@di.uoa.gr

[2]Computer Technology Institute and Press ,
N. Kazantzaki Str,
University Campus 26504 Rio Greece.
Email: schouv@di.uoa.gr

ABSTRACT

A promising approach when dealing with massive data sets is to apply randomized dimensionality reduction and then operate in lower dimensions. This paper deals with the randomized linear regression task in the case where the available data are sporadically corrupted. Instead of relying to minimization of norms, which are robust to outliers, an alternative route is taken. Building upon the observation that outliers can be detected, while operating in a low dimensional randomized projections produced embedding, a mechanism for iteratively detecting and excluding corrupted data is proposed. As a result, the linear regression is performed using conventional LS approximation, without the need to resort to linear programming-based ℓ_1 norm minimization tasks.

Index Terms— Robust Regression, Randomized algorithms

I. INTRODUCTION

Linear regression based on Least Squares (LS) approximation (known as ℓ_2 regression) has been for years a major tool for data analysis, modeling and prediction in various fields such as data mining and machine learning. However, the ever-increasing volume and complexity of the data, associated with a number of modern applications in the big data era, poses certain challenges; when the data dimension is large and/or the number of data vectors is much larger than the data dimension, then computing the exact LS solution can be quite cumbersome if not infeasible. Randomized methods for dimensionality reduction pave the way for solving efficiently and in good approximation the linear regression problem for large-scale data applications [1], [2], [3], [4]. Two are the major approaches, which have been followed so far. According to the first one, referred to as *randomized projections*, the available data are linearly combined in a randomized fashion in order to work with a lower number of vectors, which is manageable for efficient processing with the available computational and storage resources. The second one, known as *randomized sampling*, among the whole lot of data vectors, it randomly picks, usually according to a data-derived sampling distribution, a small number of them for use in the regression task. Both approaches aim at reducing the regression problem by working on a low-dimensional embedding, which approximately preserves aspects of the underline geometry, such as pairwise distances [2].

Often in practice, the available data are sporadically corrupted or hit by heavy-tailed distributed noise, rendering some of them to outliers. This can be considered as being typical in Big Data applications. In the presence of outliers, the performance of LS minimization can be severely degraded due to the large residuals appearing whenever an outlier hits. As a result, one has to resort to minimizing norms manifesting robustness to outliers, with the ℓ_1 norm being one among the most popular choices. The corresponding task is known as ℓ_1 regression or the Least Absolute Deviation problem. It is only recently that efficient methods, for providing embeddings suitable for ℓ_1 minimization, have been developed [5], [6], [7], [8]. These methods are limited to the randomized sampling approach, with the associated sampling distribution derived from a well-conditioned, with respect to the ℓ_1 norm, basis of the data subspace. After sampling, a suitable subset of data, the LAD task is solved via linear programming in the reduced dimensional space.

In this paper, an alternative route to robust randomized linear regression is taken. Building upon the observation that outliers can be detected while operating in a randomized projections produced embedding, a mechanism for iteratively detecting and excluding corrupted data is proposed. As a result, the linear regression is performed using conventional LS approximation in the reduced dimensional space, without needing to resort to linear programming-based LAD techniques. To the best of our knowledge, this is the first time that robust regression is performed directly on randomized projections–based embeddings, generated via ℓ_2-optimized fast Johnson-Lindenstrauss transforms.

Notation: Lowercase (uppercase) boldfaced letters stand for vectors (matrices). The set of real numbers is denoted by $\mathbb{R}$. Moreover, $\|\cdot\|_2$, $\|\cdot\|_1$ are the Euclidean and ℓ_1 norms respectively. Consider that the operation $\Lambda = \text{Supp}(\boldsymbol{r}, K)$, $\boldsymbol{r} \in \mathbb{R}^m$ turns Λ into an index set, subset of $\{1, \cdots, m\}$, comprising the indices of the K larger in magnitude coefficients of $\boldsymbol{r}$ and $\Lambda^c = \{1, \cdots, m\} \setminus \Lambda$. For $\Lambda \subset \{1, \cdots, m\}$, $\Delta \subset \{1, \cdots, n\}$, then $\boldsymbol{A}_{\Lambda,\Delta}$ is the submatrix of $\boldsymbol{A} \in \mathbb{R}^{m,n}$ described by the rows and columns indexed in the sets Λ and Δ respectively. If Λ or Δ is substituted by a dot $(.)$, then all the rows or all the columns are selected respectively. The same notation is applied to vectors too.

II. PROBLEM FORMULATION

The typical linear regression model is considered, i.e.,

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_* + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times l}$ is the matrix of N regressors (input vectors), $\boldsymbol{b} \in \mathbb{R}^N$ is the vector of responses and $\boldsymbol{\eta}$ the noise vector. Vector $\boldsymbol{x}_* \in \mathbb{R}^l$ comprises the unknown parameters of the linear model, which, based on least squares (LS) minimization, is estimated by

$$\boldsymbol{x}_{LS} = \arg\min_{\boldsymbol{x} \in \mathbb{R}^l} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2. \tag{2}$$

LS estimation is the most common choice in the over-determined case, i.e., $N > l$ whenever $\boldsymbol{\eta}$ is normally distributed. Assuming for simplicity that $\boldsymbol{A}$ is full rank, then the LS solution is unique and given by $\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$. Relying on, e.g., QR decomposition, the computational complexity of this problem is $\mathcal{O}(Nl^2)$, [9].

Randomized methods offer the opportunity for obtaining an approximation of $\boldsymbol{x}_{LS}$ by solving the LS problem in a reduced dimensional space:

$$\boldsymbol{x}_{\boldsymbol{R}} = \arg\min_{\boldsymbol{x} \in \mathbb{R}^l} \|\underline{\boldsymbol{b}} - \underline{\boldsymbol{A}}\boldsymbol{x}\|_2^2, \tag{3}$$

The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944 by the FET HANDICAM FP7. This work is partly supported by Marie Curie IEF, "SOL", 302898.

where $\underline{\boldsymbol{b}} = \boldsymbol{R}\boldsymbol{b}$, $\underline{\boldsymbol{A}} = \boldsymbol{R}\boldsymbol{A}$ and $\boldsymbol{R} \in \mathbb{R}^{d\times N}$ with $d \ll N$, is a carefully designed matrix, which maps the N dimensional columns of $\boldsymbol{A}$ onto a lower dimensional space of dimension d. As it will be discussed later on, $\boldsymbol{R}$ either performs randomized row sampling or randomized projection, depending on the way of its construction. The computational complexity for (3) equals to $\mathcal{O}(dl^2)+\mathcal{T}_R$, where $\mathcal{T}_R$ is the computational cost accounting for the construction and application of matrix $\boldsymbol{R}$.

Often in practice, the entries of the observed vector $\boldsymbol{b}$ are sporadically corrupted and turned, therefore, to outliers which heavily disagree with the adopted model. In this case, the noise vector can be described as $\boldsymbol{\eta} = \boldsymbol{n} + \boldsymbol{o}$ where the elements of $\boldsymbol{n}$ are normally distributed and $\boldsymbol{o}$ is a sparse vector having nonzero values only at those indices corresponding to corrupted data. In the presence of outliers, the performance of LS solution can severely degrade, due to the large residuals appearing (which are squared) whenever an outlier hits. As a result, one has to resort to cost functions manifesting robustness to outliers, with the least absolute deviations (LAD), i.e. $\boldsymbol{x}_{LAD} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^l} \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_1$ being the one adopted for randomized linear regression in [5], [6], [7], [8].

III. RANDOMIZED ALGORITHMS FOR ℓ_2 LINEAR REGRESSION

All randomization-based methods for ℓ_2 linear regression, in one way or the other, revolve around the seminal Johnson-Lindenstrauss (JL) lemma and its extensions [10], [1], which asserts that for any set of l points in $\mathbb{R}^N$ there exists a linear mapping $\mathbb{R}^N \to \mathbb{R}^d$ with $d = \mathcal{O}(\epsilon^{-2}\log l)$, so that all pairwise distances among the points are preserved up to a multiplicative factor between $(1-\epsilon)$ and $(1+\epsilon)$. In the regression problem at hand, the points in the high dimensional space, are the vector $\boldsymbol{b}$ and the columns of $\boldsymbol{A}$ and the previously mentioned linear mapping is expressed with left-multiplication by the matrix $\boldsymbol{R} \in \mathbb{R}^{d\times N}$. Such matrices, which satisfy the property above with constant probability, even with different dependencies of d, are collectively referred to as JL transforms. In other words, a JL transform embeds a few points lying in a high dimensional space onto a lower dimensional one preserving aspects of their Euclidean geometry. This is why solving (3) can give similar $\boldsymbol{x}_*$ estimates with (2).

Besides the remarkable theoretical results, in practice, the efficient employment of randomized methods requires both the computationally "cheap" construction of $\boldsymbol{R}$ and the efficient performance of the matrix multiplication $\boldsymbol{R}\boldsymbol{A}$. Note that if $\boldsymbol{R}$ is a dense random matrix and/or $\boldsymbol{R}\boldsymbol{A}$ is performed naively, then $\mathcal{O}(Nl^2)$ multiplications are required; this is of the same order to the cost asked by the original LS task in the large dimension. Next, the random sampling and random projections approaches for succeeding fast construction and evaluation of $\boldsymbol{R}$ are discussed.

III-A. Randomized Projections Approach

A lot of research effort has been invested for the construction of JL transform matrices liable to less costly operations. This includes matrices comprising randomly generated ± 1 values [11], sparse matrices, and structured matrices which involve fast transforms such as fast Fourier Transform or Walsh-Hadamard transform [12], [13], [3], [14].

A general form of a fast JL matrix is $\boldsymbol{R} = \boldsymbol{P}\boldsymbol{H}\boldsymbol{D}$, where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with ± 1 values chosen uniformly at random, $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ is a Hadamard matrix with columns normalized to unit norm and $\boldsymbol{P} \in \mathbb{R}^{d\times N}$ is a sparse matrix. In its simplest form, $\boldsymbol{P}$ is a sampling matrix; i.e., it just collects randomly d rows of the matrix $\boldsymbol{H}\boldsymbol{D}$. The multiplication of the Hadamard matrix with a vector $\boldsymbol{a} \in \mathbb{R}^N$, costs $\mathcal{O}(N\log N)$ via the fast Walsh-Hadamard transform. Moreover, if only $k < N$ components of the resulted vector $\boldsymbol{H}\boldsymbol{x}$ are needed, i.e., k is at most the number of non-zeros in $\boldsymbol{P}$, then the above complexity boils down to $\mathcal{O}(N\log k)$ [13]. Accordingly, the application of a fast JL matrix to the l columns of the matrix $\boldsymbol{A}$ costs $\mathcal{O}(lN\log k)$.

Left multiplication of $\boldsymbol{A}$ with a matrix $\boldsymbol{R}$ such as those discussed so far, leads to a matrix $\underline{\boldsymbol{A}}$ comprising a smaller number of rows with each one of them being a linear combination of the rows of the original matrix $\boldsymbol{A}$; i.e., it linear combination of the available data. This is a key characteristic of the random projections approach.

III-B. Randomized Sampling Approach

The alternative route towards a low dimensional embedding is via random sampling, according to which $\boldsymbol{R}$ is a row-sampling operator; that is, it picks d rows from $\boldsymbol{A}$ and the corresponding coefficients of $\boldsymbol{b}$ imposing, at the same time, a proper re-scaling on them [15], [2]. The aim is to mainly pick with replacement those rows of $\boldsymbol{A}$ and the corresponding components of $\boldsymbol{b}$, which are *the most influential* in determining the best LS fit. Information about the importance of each data vector is offered by the so–called *statistical leverage scores*, which are given by

$$\ell_i = \|\boldsymbol{U}_{i,.}\|_2^2, \tag{4}$$

where $\boldsymbol{U}$ is any orthonormal matrix spanning the column space of $\boldsymbol{A}$, e.g., a matrix comprising the d left singular vectors. The major random sampling algorithms use these scores (or estimates of them) to construct an importance sampling distribution $\{p_i\}_{i=1}^N$, with $p_i = \frac{\ell_i}{l}$, in order to sample the rows of $\boldsymbol{A}$ with respect to it, (see [2] for an indebt discussion on statistical leverage scores and associated sampling strategies). Intuitively, the larger the p_i is the higher the probability of randomly selecting the ith row of $\boldsymbol{A}$ becomes. In order to realize the row selection via matrix $\boldsymbol{R}$, the latter is first initialized to a zero matrix and then a certain row, say the ith one, admits a unique nonzero value as follows: An integer value, say $\rho \in [1,\dots,N]$ is randomly generated according to the sampling distribution, indicating that the ith row of the reduced matrix $\underline{\boldsymbol{A}}$ will be the ρth row of $\boldsymbol{A}$ re-rescaled by the value $1/(dp_\rho)$. In order this to happen, when left-multiplying with $\boldsymbol{R}$, its entry $\boldsymbol{R}_{i,k}$ admits the value $1/(dp_\rho)$.

Similarly to the random projections approach, the fast Walsh-Hadamard transform plays a key role in the random sampling methods as well. Indeed, the naive computation of the leverage scores, i.e. to obtain the left singular vectors via an SVD, costs as much as the LS minimization in the high dimension. Happily, leverage scores can be approximated as follows [16]:

$$\hat{\ell}_i = \|\boldsymbol{e}_i^T\boldsymbol{A}(\boldsymbol{\Pi}_1\boldsymbol{A})^\dagger\boldsymbol{\Pi}_2\|_2^2, \tag{5}$$

where $\boldsymbol{e}_i$ is a standard basis vector and the matrices $\boldsymbol{\Pi}_1 \in \mathbb{R}^{r_1\times N}$, $\boldsymbol{\Pi}_2 \in \mathbb{R}^{l\times r_2}$ are a fast JL transform, e.g. such as the one based on the Hadamard transform described before, and an ordinary JL transform, e.g. a matrix whose entries are chosen i.i.d. from a Gaussian distribution [17].

III-C. Randomized methods for ℓ_1 linear regression

As it has already been discussed, in robust regression, the ℓ_2 norm minimization is no longer suitable and one escaping route is to resort to ℓ_1 regression. However, in this case, the embedding methods which preserve Euclidean distances, are no longer valid, in theory, at least. The explanation for that is fairly simple. The length, in terms of the ℓ_1 norm, of a vector is not invariant under rotation. As a result, the leverage scores in the ℓ_1 scenario, given by $\bar{\ell}_i = \|\boldsymbol{W}_{i,.}\|_1$, $i = 1,\cdots,N$, have to be computed not from any orthonormal matrix $\boldsymbol{W}$, spanning the data subspace, but from one which is *well-conditioned* in order to preserve ℓ_1 distances. It is only very recently that results in the spirit of the JL transform were published for the ℓ_1 case. It turns out that, the corresponding transform matrix is Cauchy-distributed [5], [6]. A fast transform of this type has already been presented, [7], [8].

In practice, however, the merits of Cauchi-based ℓ_1 leverage scores estimation are observed only when N is way much larger than l [18]. As a result, it turns out that the orthonormal bases induced by methods optimized for the ℓ_2 case, e.g., [16] can also be used as well-conditioned basis for the estimation of ℓ_1 leverage scores. This is the approach used in the numerical evaluation section.

IV. ITERATIVE RANDOMIZED ROBUST REGRESSION

Let $\Lambda \subset \{1, \cdots, N\}$ be the index set indicating the data pairs which correspond to outliers. Ideally, these data pairs should have been excluded from $\boldsymbol{A}$ and $\boldsymbol{b}$ before the dimensionality reduction task. In other words, $\boldsymbol{R}\boldsymbol{A}_{\Lambda^c,:}$ and $\boldsymbol{R}\boldsymbol{b}_{\Lambda^c}$ should have been used for the LS estimation of $\boldsymbol{x}_*$ instead of $\underline{\boldsymbol{A}}$ and $\underline{\boldsymbol{b}}$. Our present work is built upon the observation that outliers can be detected while operating in the reduced dimensional space and on top of that, the corrupted data can be removed from the full dataset, without taking them back to the high dimensional space. In contrast to the already established randomized robust estimation methods, the proposed approach features two novel characteristics, which are different from any method proposed so far. First, it uses randomized projections instead of randomized sampling and second it is not relying on minimization of norms, which are robust to outliers, such as the ℓ_1 norm.

As it was mentioned above, the outlier data, once detected, can be effectively excluded from $\underline{\boldsymbol{A}}$ and $\underline{\boldsymbol{b}}$ while working in the reduced dimensional space as follows:

$$\boldsymbol{R}\boldsymbol{b}_{\Lambda^c} = \underline{\boldsymbol{b}} - \boldsymbol{R}_{:,\Lambda}\boldsymbol{b}_{\Lambda}, \tag{6}$$

$$\boldsymbol{R}\boldsymbol{A}_{\Lambda^c,:} = \underline{\boldsymbol{A}} - \boldsymbol{R}_{:,\Lambda}\boldsymbol{A}_{\Lambda,:}. \tag{7}$$

Moreover, successive application of (6), using subsets of Λ, allows the data–clearing to be performed progressively, e.g. expelling outliers one after the other or in groups in an iterative fashion.

Following the discussion in II, when the dimensionality reduction is performed in the presence of outliers, (1) is written as:

$$\underline{\boldsymbol{b}} = \underline{\boldsymbol{A}}\boldsymbol{x}_* + \underline{\boldsymbol{n}} + \boldsymbol{R}\boldsymbol{o}, \tag{8}$$

where $\underline{\boldsymbol{n}} = \boldsymbol{R}\boldsymbol{n}$. Note that if $\boldsymbol{R}$ is a random projection matrix, then $\boldsymbol{R}\boldsymbol{o}$ loose the sparsity property, that the high dimensional outlier vector has, and the energy of the nonzero values of $\boldsymbol{o}$ is spread across the d dimensions.

Observe from (8) that, the overall noise involved is given by $\underline{\boldsymbol{n}} + \boldsymbol{R}\boldsymbol{o}$. The first noise term, $\underline{\boldsymbol{n}} = \boldsymbol{R}\boldsymbol{n}$, for $\boldsymbol{R}$ being the fast JL matrix and due to the orthogonality of the Hadamard transform is normally distributed. The second one, due to the central limit theorem and unless S is trivially small, it can be assumed to be normally distributed as well. In other words, due to Gaussianity, $\boldsymbol{R}$ renders the problem suitable for LS minimization in the reduced dimension. However, the noise variance is proportional to the number of outliers and the magnitude of the outlier vector components. Accordingly, the LS minimization in the reduced dimensional can provide a tentative estimate $\hat{\boldsymbol{x}}$, i.e.

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\tilde{\underline{\boldsymbol{b}}} - \underline{\boldsymbol{A}}\boldsymbol{x}\|_2^2, \tag{9}$$

where $\tilde{\underline{\boldsymbol{b}}} = \underline{\boldsymbol{b}} - \underline{\boldsymbol{n}} - \boldsymbol{R}\boldsymbol{o}$ is the vector of the noisy observations.

We next turn our attention to ways for detecting data vectors which correspond to outliers. After replacing $\boldsymbol{x}_*$ with the tentative estimate, (8), yields

$$\boldsymbol{z} = \boldsymbol{R}\boldsymbol{o}, \tag{10}$$

where $\boldsymbol{z} = \underline{\boldsymbol{b}} - \underline{\boldsymbol{A}}\hat{\boldsymbol{x}} - \underline{\boldsymbol{n}}$. This is the typical problem that compressed sensing with inaccurate measurements is dealt with, with $\boldsymbol{z}$ being the noisy observed measurements, $\boldsymbol{o}$ being the unknown sparse vector and $\boldsymbol{R}$ playing the role of the sensing matrix [19]. Denote the error vector corresponding to the tentative LS solution as $\boldsymbol{x}_e = \hat{\boldsymbol{x}} - \boldsymbol{x}_*$, then, $\underline{\boldsymbol{A}}\hat{\boldsymbol{x}} = \underline{\boldsymbol{A}}\boldsymbol{x}_* + \underline{\boldsymbol{A}}\boldsymbol{x}_e$, where the second term contributes to the overall noise. The recovery accuracy of $\boldsymbol{o}$ depends on its sparsity level, i.e. $S = \|\boldsymbol{o}\|_0$, on some Restricted Isometry Property (RIP) constant of $\boldsymbol{R}$ and $\|\underline{\boldsymbol{n}}'\|_2$, where $\underline{\boldsymbol{n}}'$ denotes the overall noise. [19], [20], [21].

Table I. Iterative Randomized Robust Linear Regression (IRRLR)

Set Parameters: K, d, I
Generate matrix $\boldsymbol{R} = \boldsymbol{P}\boldsymbol{H}\boldsymbol{D}$
Compute $\underline{\boldsymbol{A}}^{[0]} = \boldsymbol{R}\boldsymbol{A}$,
and $\underline{\boldsymbol{b}}^{[0]} = \boldsymbol{R}\boldsymbol{b}$
FOR $i = 1, 2, \ldots, I$
Step 1: Estimate $\hat{\boldsymbol{x}}$ via (9)
Step 2: Compute residual, $\boldsymbol{z} = \underline{\boldsymbol{b}} - \underline{\boldsymbol{A}}\hat{\boldsymbol{x}}$
Step 3: Compute proxy as $\boldsymbol{\psi} = \boldsymbol{R}^T\boldsymbol{z}$
or as $\boldsymbol{\psi} = \hat{\boldsymbol{o}}$ via (10)
Step 4: Set $\Lambda = \mathrm{Supp}(|\boldsymbol{\psi}|,\ K)$
Step 5: Compute $\underline{\boldsymbol{A}}^{[i]} = \underline{\boldsymbol{A}}^{[i-1]} - \boldsymbol{R}_{:,\Lambda}\boldsymbol{A}_{\Lambda,:}$
and $\underline{\boldsymbol{b}}^{[i]} = \underline{\boldsymbol{b}}^{[i-1]} - \boldsymbol{R}_{:,\Lambda}\boldsymbol{b}_{\Lambda}$
ENDFOR

We propose a technique for progressively clearing up the available data from outliers which is iterative in nature and the reasoning behind it is explained next: In the presence of large/many outliers, the error term $\boldsymbol{x}_e$ resulting from $\hat{\boldsymbol{x}}$ is expected to take relatively large values. This will lead to increased noise in (10) affecting the estimation accuracy of $\boldsymbol{o}$. However, the good news is that, in principle, the proposed method does not require an accurate estimate of $\boldsymbol{o}$. It is good enough just to detect *a subset* or even a single element of the support of $\boldsymbol{o}$. If this is succeeded, then the corresponding outlier data can be omitted from $\underline{\boldsymbol{A}}$ and $\underline{\boldsymbol{b}}$ with the aid of (6). The same procedure is iterated for a number of times in order to detect more subsets of the set of the remaining outliers, which are used for further cleaning the available data.

The immunity of the proposed scheme against outliers is enhanced due to the following reasons: First, it is true that due to inaccuracies in the estimates of $\hat{\boldsymbol{x}}$ and $\boldsymbol{o}$, it is possible a number of data to be characterized as outliers without really being so. However, in big data applications, where a very large pool of data is available to work with, this is not of major concern. As long as outliers are detected and excluded from further computations, then in principle, it is not harmful to expel some healthy data as well. Second, after the completion of each iteration step, the processed data are likely to be healthier since they have been purified from an increased number of outliers. This leads to improved estimates of $\hat{\boldsymbol{x}}$ and, hence, it is likely to get more accurate outlier detection from one iteration to the next.

Because the algorithm does not need a full estimate of $\boldsymbol{o}$, the solution of the compressed sensing problem in (10) can be simplified for computational complexity savings. For this reason, estimates of subsets of the support of $\boldsymbol{o}$ can be obtained using as proxy the product $\boldsymbol{\psi} = \boldsymbol{R}^T\boldsymbol{z}$ and then $\Lambda = \mathrm{Supp}(\boldsymbol{\psi}, K)$, where $K \leq N$. This approach appears in the first iteration step of greedy algorithm such as CoSaMP [22]. The algorithmic steps of the method, hereafter referred to as Iterative Randomized Robust Linear Regression (IRRLR), are described in Table I.

V. COMPUTATIONAL COMPLEXITY ANALYSIS

According to Table I, the required computational complexity is analyzed as follows: The computation of $\underline{\boldsymbol{A}}^{[0]}$ and $\underline{\boldsymbol{b}}^{[0]}$ costs $\mathcal{O}((l+1)N\log k)$, where k is the number of nonzero values of matrix P, and it is performed once. The rest of the computations, corresponding the the steps of the algorithm, are performed I times and their complexity is: For step 1, the LS task in the reduced dimension takes $\mathcal{O}(dl^2)$ and an other $d(l+1)$ is needed for step 2. Step 3, is rewritten as $\boldsymbol{R}^T\boldsymbol{r} = \boldsymbol{D}\boldsymbol{H}\boldsymbol{P}^T\boldsymbol{r}$, where $\boldsymbol{P}^T\boldsymbol{r}$ costs $\mathcal{O}(k)$ and its Hadamard transform, performed via direct multiplication,

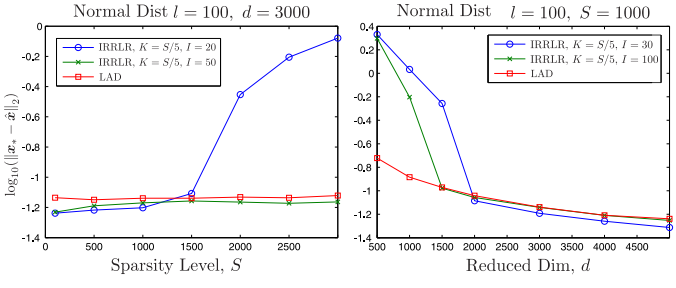

Fig. 1. Performance evaluation (Gaussian $\boldsymbol{A}$)

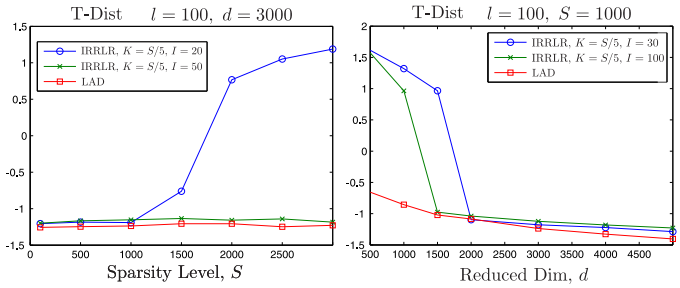

Fig. 2. Performance evaluation (T-Distributed $\boldsymbol{A}$)

costs an extra $\mathcal{O}(Nk)$. The evaluation of the diagonal matrix $\boldsymbol{D}$ just causes at most N sign changes and it is not counted. For step 4, any selection algorithm of complexity $\mathcal{O}(N)$ can be employed for the detection of the Kth larger component and then a single run across the vector returns the rest of required indices. Finally, step 5, in case that $\boldsymbol{R}$ is pre-computed and stored, it needs $\mathcal{O}(dKl)$ or an extra $\mathcal{O}(kK)$ avoiding storage.

With respect to the random sampling using leverage scores approach, the approximate leverage scores in (5) can be computed in $\mathcal{O}((l+1)N\log r_1 + lNr_2 + r_1 l^2 + r_2 l^2)$, [16], where for comparison purposes r_1 can be considered to be equal to d and r_2 can be set equal to $\mathcal{O}(\log l)$. Apart from that, large computational expenses result from the LAD optimization, which does not admit a closed form solution. As a result, one has to either resort to linear programming using, e.g. interior-point methods with complexity poly(d). Approximate solution with lower complexity could also be a choice, e.g., the ADMM approach in [23].

VI. NUMERICAL EXAMPLES

In this section the performance of the proposed method in investigated with the aid of synthetic numerical examples. First, an example of $l = 100$, with $N = 2^{17}$ data vectors in total and $\boldsymbol{A} \in \mathbb{R}^{2^{17}\times 100}$ Gaussian distributed is adopted in order to examine the capacity of IRRLR in detecting outliers when operating in a reduced dimensionality equal to $d = 3000$. Additive Gaussian noise is added accounted of 30dB SNR and $S = 2000$ outliers are also imposed in randomly picked positions. Their values are randomly drawn from a Gaussian distribution $\mathcal{N}(0,1) * \mu$, where μ is chosen to be equal to the larger in magnitude value of $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_*$. This guarantees that most of the outliers will not pop up over the noise floor rending them easily to withdraw via direct thresholding.

The results are shown in Fig. 3, where the figure on top shows the total number of outliers detected as a function of the number of iterations, where in each iteration $S/5$ data vectors are excluded for further processing. The figure in bottom shows the corresponding error in terms of $\log_{10}(\|\boldsymbol{x}_* - \hat{\boldsymbol{x}}\|_2)$. Observe that, around 30 iterations are needed in order to clean most of outliers and reach a performance error floor. Note that, in all simulation examples matrix $\boldsymbol{P}$ is assigned only a single nonzero value per row, i.e., $k = d$.

Next, the performance of IRRLR is evaluated against the LAD estimation where dimensionality reduction is realized via random sampling based on the approximate leverage scores as it was described above and according to [16], [18]. In this case, LAD is minimized with an interior point method. The results are shown in Fig. 1 and Fig. 1 and they correspond to Gaussian- and T-distributed input vectors respectively. The order of the T-Distribution was set equal to 2. Small order values is known to produce highly irregular leverage scores, [24].

In Fig. 1, left, the reduced dimension is kept fixed and equal to $d = 3000$ and displays the performance for different number of outliers. In each iteration $S/5$ data vectors are considered to be outliers and they are expelled from $\underline{\boldsymbol{R}}$ and $\underline{\boldsymbol{b}}$. It turns out that, when an adequate number of iterations is used (in particular $I = 50$), then IRRLR outperforms LAD for all the evaluated range of outliers number. Note that this is realized with a lower computational burden. The curve denoted by circles shows the behaviour of the method when a number of iterations is large enough for clearing up to $S = 1000$ outliers. In Fig. 1, right, the reduced dimension is varying taking values from 500 up to 5000, whereas the number of outliers is kept fixed and equal to $S = 1000$. It is observed that when d is getting as small as the number of outliers, which are present, then IRRLR fails to perform as good as the randomized sampling LAD, at least performing a moderate number of iterations. On the contrary, in the rest of the cases, i.e. for $d > 1500$ it performs somewhat better.

When $\boldsymbol{A}$ is T-distributed with order 2, then the results are similar with the difference that LAD is somewhat better than IRRLR in all tested configurations as it is shown in Fig. 2. This is expected, since leverage scores estimates are particularly suited to detect and favor the most significant data vectors, whereas IRRLR in its current form treats all data equivalently. In any case, it should be accounted the fact the IRRLR exhibits much faster running times.

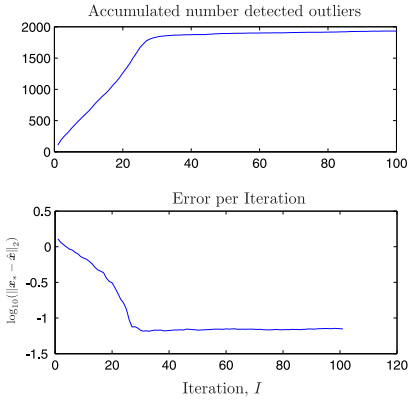

Fig. 3. Capability of IRRLR to accumulatively detect outliers

VII. CONCLUSIONS AND FUTURE WORK

The approach for robust randomized linear regression proposed here departs from recently proposed state-of-the-art randomized sampling algorithms, which are based on leverage scores. The presented preliminary results show that similar performance for a wide range of configurations can be achieved without employing computationally heavy linear programming techniques. The method was kept as simple as possible and it serves as a proof of concept. Following the same rationale, improvements can be made in several stages of the algorithm and are left for future work. For example, more advanced compressed sensing approaches can be employed for partial support estimation. Moreover, it is flexible to exploit a priori information about the characteristics of the outliers. For example, in many applications, outliers are appearing in bursts. In such scenarios block sparse estimation is expected to enhance performance.

VIII. REFERENCES

[1] T. Sarlós, "Improved approximation algorithms for large matrices via random projections," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 2006, pp. 143–152.

[2] M. W. Mahoney, "Randomized algorithms for matrices and data," *Found. Trends Mach. Learn.*, vol. 3, no. 2, pp. 123–224, Feb. 2011.

[3] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, "Faster least squares approximation," *Numerische Mathematik*, vol. 117, no. 2, pp. 219–249, 2011.

[4] N. Ailon and E. Liberty, "An almost optimal unrestricted fast johnson-lindenstrauss transform," *ACM Transactions on Algorithms*, vol. 9, no. 3, pp. 1–12, Jun. 2013.

[5] A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney, "Sampling algorithms and coresets for ℓ_p regression," *SIAM Journal on Computing*, vol. 38, no. 5, pp. 2060–2078, Jan. 2009.

[6] C. Sohler and D. P. Woodruff, "Subspace embeddings for the l 1-norm with applications," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, 2011, pp. 755–764. [Online]. Available: http://dl.acm.org/citation.cfm?id=1993736

[7] X. Meng and M. W. Mahoney, "Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression," in *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '13. New York, NY, USA: ACM, 2013, pp. 91–100.

[8] K. L. Clarkson, P. Drineas, M. Magdon-Ismail, M. W. Mahoney, X. Meng, and D. P. Woodruff, "The fast cauchy transform and faster robust linear regression," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2013, pp. 466–477.

[9] G. H. Golub and C. F. van Van Loan, "Matrix computations (johns hopkins studies in mathematical sciences)," 1996.

[10] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in *Conference in modern analysis and probability (New Haven, Conn., 1982)*, ser. Contemporary Mathematics. American Mathematical Society, 1984, vol. 26, pp. 189–206.

[11] D. Achlioptas and F. Mcsherry, "Fast computation of low-rank matrix approximations," *Journal of the ACM (JACM)*, vol. 54, no. 2, p. 9, 2007.

[12] N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast johnson-lindenstrauss transform," in *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM, 2006, pp. 557–563.

[13] N. Ailon and E. Liberty, "Fast dimension reduction using rademacher series on dual BCH codes," *Discrete & Computational Geometry*, vol. 42, no. 4, pp. 615–630, Dec. 2009.

[14] J. A. TROPP, "Improved analysis of the subsampled randomized hadamard transform," *Advances in Adaptive Data Analysis*, vol. 03, no. 01–02, pp. 115–126, 2011.

[15] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Sampling algorithms for l2 regression and applications," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, ser. SODA '06. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006, pp. 1127–1136.

[16] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, "Fast approximation of matrix coherence and statistical leverage," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3475–3506, 2012.

[17] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[18] K. L. Clarkson, P. Drineas, M. Magdon-Ismail, M. W. Mahoney, X. Meng, and D. P. Woodruff, "The fast cauchy transform: with applications to basis construction, regression, and subspace approximation in L1," *CoRR*, vol. abs/1207.4684, 2014.

[19] E. J. Candès, J. Romberg, and T. Tao, "Stable recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[20] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, April 2010.

[21] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, Jul. 2011.

[22] D. Needell and J. A. Tropp, "COSAMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[24] G. Raskutti and M. Mahoney, "A statistical perspective on randomized sketching for ordinary least-squares," *arXiv preprint arXiv:1406.5986*, 2014. [Online]. Available: http://arxiv.org/abs/1406.5986