GLIMPS: A Greedy Mixed-Integer Approach for Super Robust Matched Subspace Detection

Md Mahfuzur Rahman[1], Daniel Pimentel-Alarcón[2]
[1]Georgia State University, [2]University of Wisconsin-Madison

Abstract—**Due to diverse nature of data acquisition and modern applications, many contemporary problems involve high dimensional datum $\mathbf{x} \in \mathbb{R}^d$ whose entries often lie in a union of subspaces and the goal is to find out which entries of $\mathbf{x}$ match with a particular subspace $\mathbb{U}$, classically called** *matched subspace detection*. **Consequently, entries that match with one subspace are considered as inliers w.r.t the subspace while all other entries are considered as outliers. Proportion of outliers relative to each subspace varies based on the degree of coordinates from subspaces. This problem is a combinatorial NP-hard in nature and has been immensely studied in recent years. Existing approaches can solve the problem when outliers are sparse. However, if outliers are abundant or in other words if $\mathbf{x}$ contains coordinates from a fair amount of subspaces, this problem can't be solved with acceptable accuracy or within a reasonable amount of time. This paper proposes a two-stage approach called** *Greedy Linear Integer Mixed Programmed Selector* **(GLIMPS) for this abundant-outliers setting, which combines a greedy algorithm and mixed integer formulation and can tolerate over 80% outliers, outperforming the state-of-the-art.**

I. Introduction

Approaches for data analysis has been greatly advanced in recent years. It results in the emergence of high-dimensional data in all areas of science, business, and engineering problems. To this end, high-dimensional data can often be best explained using low-dimensional linear subspaces. In this model, a high dimensional datum $\mathbf{x} \in \mathbb{R}^d$ is a point lying in a subspace $\mathbb{U}$. The task of testing whether a given datum $\mathbf{x}$ lies in the given subspace $\mathbb{U}$ or not, is commonly known as *matched subspace detection* (MSD) [1]. Testing whether $\mathbf{x}$ is a complete inlier vector or not can be readily accomplished using the mathematical relationships between a vector $\mathbf{x}$ and its corresponding subspace $\mathbb{U}$. The classical formulation of this MSD problem is formulated as a hypothesis test problem for which optimality of the solution is guaranteed based on the amount of relative signal energy of $\mathbf{x}$ w.r.t. the given subspace $\mathbb{U}$ [1], [2]. Different variants of this *matched subspace detection* problem has numerous applications in literature including remote sensing systems [3], [4], neural activation detection [5], anomaly detection [6], [7], and communications [8]. In those applications, it is assumed that outliers are sparse in datum. However, many contemporary applications and its variants including computer vision [9], hyperspectral imaging [10], recommender systems [11], [12], lidar [13], and many more inherently may allow vast majority of outliers in problem setting. Furthermore, several other

factors (e.g. missingness , inconsistencies, noises and outliers in data) may also come into picture to make the problem very difficult to solve. To cope with these varied situations, a several other modified formulations of MSD [14]–[16] are proposed in the literature. For instance, it may be often the case that full measurement of $\mathbf{x}$ is not feasible or not allowed as assumed in [17], [18].

In this paper, we will assume two variants of MSD formulation, namely *robust matched subspace detection* (RMSD) and *super robust matched subspace detection* (SRMSD). By RMSD, we mean a few number of coordinates of $\mathbf{x}$ ($< \frac{d}{2}$) is not just slightly perturbed by small noises. Instead, those entries, by any means, have been replaced completely by wrong values (outliers). SRMSD refers to the case that the most of the coordinates ($>> \frac{d}{2}$) of the given vector $\mathbf{x}$ has been severely corrupted. In both cases, the goal is to identify the inlier entries in $\mathbf{x}$.

II. Related Work

The RMSD variant of the classical MSD problem can be easily tackled using the ℓ_1-norm minimization of the residual $\mathbf{x} - \mathbf{U}\theta$ because the solution always favors majority entries which is the case in RMSD (number of inlier entries $> \frac{d}{2}$). This formulation is closely related to LASSO [19] as it also uses ℓ_1-loss as regularization parameter. This RMSD setting has been well studied in practice because numerous applications can be modeled as RMSD problems [20]–[27]. However, RMSD frameworks are not easily extensible for the problems to be modeled as SRMSD. The MSD or RMSD problem is often approached using randomized methods like [28]–[30] but it is not applicable in SRMSD setting because of its combinatorial NP-hard nature. Moreover, the solutions provided by randomized methods are not guaranteed to be globally optimal [31]–[38] even if it is an RMSD problem. However, a very recent bi-convex programming based method is proposed in [35] that aims to refine the result obtained from RANSAC using a second-stage bisection search in the remaining solution space. However, RANSAC itself is sub-optimal in solving an SRMSD problem. Another variant of RANSAC is recently proposed in [39] that ranks measurements in each iteration. Furthermore, it necessitates some parameter configurations that makes the approach harder to implement in practice. A deterministic and locally convergent method proposed in [37] can handle a large proportion of outliers but it is slow and heavily dependent on

RANSAC or Least-Squares based initial set of solutions.

In quest of global optimality for SRMSD class of problems, several approaches have been proposed in recent years. For example in MaxFS [31], similar to SRMSD problem is formulated as a set of infeasible constraints and the goal is to find the maximum feasible subset from the set. It uses deterministic branch-and-bound (BnB) methods to find the solution but for large-scale problems, unfortunately it may take exponential time [34]. Moreover, MaxFS only guarantees solution for small-scale homogenous linear systems. However, linear matrix inequality constraints within BnB method is proposed in [40] to obtain reduced time requirement and handle large amount of outliers, but it requires some pre-defined structure in the objective variables. Recently, iterative re-weighted ℓ_1-minimization method proposed in [32] can find comparatively large maximal inlier set but it requires iteratively solving a number of similar sub-problems causing it slower than RANSAC class of algorithms. In addition, it is not supportive for SRMSD formulation as its performance is highly determined by the proportion of outliers. Another tree-based technique is recently proposed in [34] to solve the problem in reduced amount of time using application dependent heuristics and based on certain pre-assumption that residual structure is to be quasi-convex. Unfortunately, in many situations, these heuristics may not be available and the assumption on the residual structure may not be accurate in practice.

An MILP-based outlier removal technique, called GORE, is proposed in [33] but it can only be used as a preprocessing routine for the class of algorithms solving maximum consensus set problem. However, its removal is guaranteed up to 20% and can't be worthwhile if outliers are abundant. More recently, a heuristic greedy algorithm is proposed in [41] for subspace tracking that aims to find the most probable outlier entry testing whether removal of this entry minimizes the gap between $\mathbf{x}$ and its projection onto $\mathbb{U}$. It is observed that this greedy algorithm, which runs in polynomial time $\mathcal{O}(\mathrm{d}^2)$, outperforms the method based on ℓ_1-minimization of the residual and tolerates up to 60% outliers.

In this paper, we propose GLIMPS which is a joint effort of the greedy [41] and MILP algorithms to utilize the power of a fast (greedy) and an exact (MILP) algorithm to solve an SRMSD problem. We show that GLIMPS can outperform other existing approaches including ℓ_1-minimization, the greedy and MILP algorithms. We also show that ℓ_1-minimization even when assisted with the greedy algorithm can't perform very well.

III. CONTRIBUTIONS OF THIS PAPER

The contribution of this paper can be summarized as follows:
- GLIMPS offers a two-stage approach that exacts the bests of both greedy and MILP algorithms. To make it more precise, greedy algorithm is fast and its performance in handling outliers is better than other randomized and ℓ_1-based approaches. However, if outliers are abundant (80% or more), then in very few cases, it unexpectedly removes some inliers as outliers. This unintended phenomenon

makes greedy algorithm failed in majority of trials if $\mathbf{x}$ is severely outlier-dominant. On the other hand, MILP is an exact algorithm, however because of the combinatorial nature of SRMSD problem, it may take exponentially long time to solve the problem. In contrast, GLIMPS reduces the search space for MILP by removing a certain percentage of most harmful entries using greedy algorithm and thus eventually makes MILP faster.
- By our proposed greedy algorithm in the first stage, we are providing means to warm-start MILP and thus scalability of MILP has been noticeably increased in GLIMPS.
- Greedy algorithm and MILP can tolerate up to 60% and 76% outliers respectively when they are separately used. On the other hand, GLIMPS offers tolerance up to 80% outliers.
- Existing approaches like ℓ_1-minimization [18], [20], [42]–[44] assume subspaces of specific coherence and uniformity of outlier locations in $\mathbf{x}$. In contrast, GLIMPS has no restriction on subspace structure or outlier locations.
- SRMSD is a generalized version for a class of problems. So, the idea of GLIMPS can easily be applied for other tasks if we can use subspace based model for the problems. Examples include robust PCA (principal component analysis), background separation, subspace tracking [18], [20], [27], [41], and matrix completion [42]–[51]. More details are discussed in Section IV.
- MILP can be implemented with a few lines of code using high-level mathematical optimization languages like AMPL [52] and JUMP [53]. A variety of solvers like GUROBI [54] and CPLEX [55] are readily available for academic uses.

IV. MOTIVATING APPLICATIONS

MSD is the classical setting of RMSD and SRMSD. Moreover, the setting of RMSD and SRMSD can be used to model many applications as mentioned in Section I and II. In fact, the idea of SRMSD can be applied when subspace-based modeling is feasible. In this section, some applications of SRMSD in recommender systems, network inference, signal processing, computer vision, and metagenomics are briefly described.

Recommender Systems. Organizations like Netflix, YouTube, Amazon and Facebook, based on their requirements, use recommender systems for video, music, product and content recommendations. Different approaches including k-NN, Pearson correlation coefficient and ensemble methods [12] have been proposed to solve the problem. However, as the user preference vector $\mathbf{x}$ usually lies in a low dimensional subspace, low-rank matrix completion is considered as potentially the best model whatsoever [42]–[47]. Unfortunately, if a single account is shared by multiple users, then the preference vector $\mathbf{x}$ may contain entries that belong to different subspaces. This situation can easily be modeled as an SRMSD problem due to predominance of outliers with respect to each subspace.

Networks Inference. According to system biology, understanding a system can be accomplished establishing

correlations among its components [56]. This idea can be applied to understand a broad class of networks (internet, biological networks, social networks, and many more) [57], [58]. In general, these interactions among components can be analyzed using mathematical graph theory, where, for example in biological network, each node represents an entity like DNA, RNA, protein and small molecules. Edges may represent activation levels or weights referring to confidence levels, strength, and reaction speeds. This network can be decomposed into multiple strongly connected components which we may refer to as subnets. In this setting, each node in subnet k, can be encoded as a datum $\mathbf{x}$ which , in turn, can be represented as a linear combination of two levels of interactions: 1) interaction to its own subnet k and 2) interaction between subnet k and other subnets. According to this modeling, $\mathbf{x}$ lies in a subspace $\mathbb{U}$. Because of the dynamic behavior of interactions, the state of each node may change dynamically over time and consequently $\mathbf{x}$ can have measurements from different subspaces that results in majority of outliers with respect to each subspace and hence we can leverage SRMSD algorithms for these class of problems.

Signal Processing In signal processing, for detection problems, usually a time-series data is given and the task is to identify how the data was actually generated. That is, whether the given vector $\mathbf{x}$ entirely consists of noises or outliers or it lies in some subspace containing the spectral signature of the signal. This *matched subspace detector* is the generalization of a special case called *matched signal detector* also called the *matched filter* which is considered as an important building block of signal processing [1]. In modern signal processing applications, e.g. target detection or anomaly detection is treated as a binary classifier that labels pixels of input images as pixels of interest or as anomalous pixels. In this detection problem, a fixed background can be treated as a data point lying in a low dimensional linear space and targeted or anomalous pixels can be treated as outliers. Based on the position and orientation of the cameras or receiving stations, it may have different amount of outliers. This idea can also be applied to identify peculiarities in solar images and thus infer about any potential events.

Computer Vision. Computer vision, as a sub-field of signal processing, often tries to segment background from the foreground to detect events occurring in a video. Though the background in traffic or video surveillance problems is subject to minor variation with environmental or instrumental changes, it can still be modeled as a low-dimensional subspace. For this kind of formulation, each video frame (an image) can be vectorized as $\mathbf{x}$ and background can be approximated as a subspace $\mathbb{U}$. The task is to find which entries in $\mathbf{x}$ match with $\mathbb{U}$ (inliers) and which entries don't match (outliers). Depending on the position and orientation of cameras and involving objects, frames i.e., $\mathbf{x}$ can have different number of outliers. If foreground objects are far from cameras, the outliers become sparse in $\mathbf{x}$ as assumed in RMSD , robust PCA [22]–[27] and other subspace tracking approaches [18], [27], [41]. However, in some data acquisition situations, this is always not the case and number of outliers can be a dominant majority. It turns out

that we need to precisely handle the situation of SRMSD. This kind of modeling can also be applicable for image inpainting [59], scene reconstruction, designing industrial robots and many more.

Metagenomics. To better understand different microorganisms present in an environmental sample, traditionally microbiologists used cultivation-based methods which in fact replicate certain genes to create biodiversity. Unfortunately, this approach is less effective in the sense that it always misses most of the microbial diversity. Recently, metagenomics, a recent field of microbial study, aims to provide means to understand all of the genomes from an environmental DNA sample, which is a mixture of genes from multiple organisms. Each genome is a collection of genes and representative of one organism. So, the task is to find gene-wise identities of genome present in the mixed sample. Metagenomics is supposed to advance knowledge and has vast impact on practical applications in the areas of medicine, agriculture and engineering. For example, in agriculture, better understanding the relationships between plants and microbes can reduce the risk of diseases in crops and livestock, provide enhanced means for adaptive farming [60]. Metagenomic problem can easily be modeled as an SRMSD problem where $\mathbf{x}$ refers to the DNA sample collected from the environment, i.e., each gene will correspond to one entry in $\mathbf{x}$. Each genome can be modeled as a subspace and the goal is to find out which gene (coordinate) corresponds to which genome (subspace). As the sample $\mathbf{x}$ contains entries (genes) from multiple organisms, there will be few inliers (correct genes) for each subspace. This formulation perfectly matches with our SRMSD setting.

V. PROBLEM FORMULATION

The problem is formulated as follows: consider an r-dimensional subspace $\mathbb{U}$ and a vector $\mathbf{x}$ in the ambient dimension $\mathbb{R}^d$. The given vector $\mathbf{x}$ contains some coordinates correct relative to the subspace $\mathbb{U}$ and the rest are all outliers relative to $\mathbb{U}$. SRMSD is defined as finding set of indices $\boldsymbol{\omega}$ of all the inlier coordinates in $\mathbf{x}$, where $\boldsymbol{\omega}$ means the largest $\boldsymbol{v} \subset [d] := \{1, \ldots, d\}$ such that $\mathbf{x}_{\boldsymbol{v}}$ lies in the subspace $\mathbb{U}_{\boldsymbol{v}}$. The subscript $\boldsymbol{v}$ refers to the restriction to the coordinates indicated in $\boldsymbol{v}$. The subspace $\mathbb{U}$ is spanned by $\mathbf{U} \in \mathbb{R}^{d \times r}$, whereas $\mathbb{U}_{\boldsymbol{v}}$ is spanned by $\mathbf{U}_{\boldsymbol{v}} \in \mathbb{R}^{|\boldsymbol{v}| \times r}$. In other words, we will discard all the coordinates except $\boldsymbol{v}$. If $\mathbf{x}_{\boldsymbol{v}}$ lies in $\mathbb{U}_{\boldsymbol{v}}$, it implies the coordinates of $\mathbf{x}$ indicated in $\boldsymbol{v}$ corresponds to the subspace $\mathbb{U}$. Consequently, the coordinates in $\mathbf{x}_{\boldsymbol{\omega}}$ are considered as inliers and the rest are considered as outliers.

Example 1. *Consider* $d = 5$ *and* $r = 2$
Given that

$$\mathbf{U} = \begin{bmatrix} 1 & 0 \\ 3 & 2 \\ 5 & 4 \\ 7 & 6 \\ 9 & 8 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 4 \\ 14 \\ 0 \\ 34 \\ 44 \end{bmatrix},$$

and according to formulation, $\mathbb{U} = \mathrm{span}\{\mathbf{U}\}$.

It is evident that $\boldsymbol{\omega} = \{2, 4, 5\}$, because

$$\mathbf{x}_{\boldsymbol{\omega}} = \begin{bmatrix} 14 \\ 34 \\ 44 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 7 & 6 \\ 9 & 8 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} = \mathbf{U}_{\boldsymbol{\omega}} \boldsymbol{\theta},$$

and because $\boldsymbol{\omega}$ is the largest $\boldsymbol{v}$ for which $\mathbf{x}_{\boldsymbol{v}}$ lies in $\mathbb{U}_{\boldsymbol{v}}$.

The fact that $\boldsymbol{\theta}$ we derived in the previous example is correct and $\mathbf{x}_{\boldsymbol{\omega}}$ are really inliers can be easily justified as follows: Given that the dimension of the subspace is r = 2 and consequently, the dimension of the coefficient vector $\boldsymbol{\theta}$ is 2. Hence, as we know $\mathbf{x} = \mathbf{U}\boldsymbol{\theta}$, we have two unknowns in the chosen system of equations $\mathbf{x}_{\boldsymbol{\omega}} = \mathbf{U}_{\boldsymbol{\omega}}\boldsymbol{\theta}$. Now, if we choose $m = |\boldsymbol{\omega}| \leq $ r equations, the system is either underdetermined or uniquely determined, and we must get a solution for $\boldsymbol{\theta}$. For example, if we choose $\boldsymbol{\omega} = \{1, 3\}$, we have a solution $\boldsymbol{\theta} = [4 \ -5]^{\mathsf{T}}$. Same is the case if we choose even fewer equations. However, this solution is not guaranteed because the system is under-constrained or uniquely constrained. To guarantee that the set of indices $\boldsymbol{\omega}$ and the corresponding inliers are correct, we need to find a solution for an overdetermined system, i.e. we need at least r + 1 equations.

VI. GREEDY LINEAR INTEGER MIXED PROGRAMMED SELECTOR

As we have mentioned in Related Work section, to address the problem in question, i.e. to solve SRMSD problem, greedy algorithm and mixed-integer linear program (MILP) based formulation have been proposed in the literature. However, if the fraction of outlier becomes huge, these techniques when used separately either fail to give correct results or take exponentially longer time. Our two-stage approach to address SRMSD is a combination of a greedy algorithm and a MILP to improve the existing results. In its first stage, we iteratively remove those coordinates (one per each iteration) whose removal minimizes the gap between $\mathbf{x}$ and its projection onto the subspace $\mathbb{U}$. Projections are computed dynamically as we restrict coordinates only to the existing coordinates. Then, in the second stage, we apply a MILP to identify the remaining inliers and outliers. Once we identify the remaining inliers, we obtain the coefficient of the inliers of $\mathbf{x}$, $\boldsymbol{\theta}$ which in turn allows us to identify *all* the inliers and outliers in $\mathbf{x}$.

First Stage: Greedy Algorithm. Suppose $\boldsymbol{\omega}^c$ is a set of coordinates of the given vector $\mathbf{x}$ that are already found to be outliers. We compute d $- |\boldsymbol{\omega}^c|$ pair (for $\mathbf{U}$ and $\mathbf{x}$) of projections onto each set of the coordinates $\{1, \ldots, d\} \setminus \{\boldsymbol{\omega}^c \cup \{i\}\}, \forall i \in \{1, \ldots, d\} \setminus \boldsymbol{\omega}^c$. Let $\boldsymbol{\omega}_i = \{1, \ldots, d\} \setminus \{\boldsymbol{\omega}^c \cup \{i\}\}$. Mathematically, $\mathbf{U}_{\boldsymbol{\omega}_i} = P_{\boldsymbol{\omega}_i}\mathbf{U}$ and $\mathbf{x}_{\boldsymbol{\omega}_i} = P_{\boldsymbol{\omega}_i}\mathbf{x}$, where $P_{\boldsymbol{\omega}_i}$ is the projection operator when $\mathbf{x}$ and $\mathbf{U}$ are restricted to the set of coordinates $\boldsymbol{\omega}_i$. Then, we compute projection of $\mathbf{x}_{\boldsymbol{\omega}_i}$ onto the projected subspace $\mathbf{U}_{\boldsymbol{\omega}_i}$, i.e., $\hat{\mathbf{x}}_{\boldsymbol{\omega}_i} = \mathbf{U}_{\boldsymbol{\omega}_i}(\mathbf{U}_{\boldsymbol{\omega}_i}^{\mathsf{T}}\mathbf{U}_{\boldsymbol{\omega}_i})^{-1}\mathbf{U}_{\boldsymbol{\omega}_i}^{\mathsf{T}}\mathbf{x}_{\boldsymbol{\omega}_i}$. For each $\boldsymbol{\omega}_i$ as defined earlier, we compute the ratio of two norms, $r_i = \frac{\|\hat{\mathbf{x}}_{\boldsymbol{\omega}_i}\|}{\|\mathbf{x}_{\boldsymbol{\omega}_i}\|} \leq 1$. It is known that larger r_i implies $\mathbf{x}_{\boldsymbol{\omega}_i}$ and $\hat{\mathbf{x}}_{\boldsymbol{\omega}_i}$ are closer. Intuitively, it implies if we remove i-th coordinate along with other $\boldsymbol{\omega}^c$ coordinates, $\mathbf{x}$ gets closer to the subspace $\mathbb{U}$ whose basis is $\mathbf{U}$. Because of having a clear set of

intuitive choices, this approach is called greedy erasure (outlier removal) algorithm. We identify the coordinate with largest ratio, and the coordinate is removed as an outlier adding the index to the outlier set of indices $\boldsymbol{\omega}^c$. Obviously, at the beginning of the algorithm, $\boldsymbol{\omega}^c = \emptyset$. We iterate this process until we have sufficient number of inliers compared to the number of outliers so that MILP can succeed in the second stage. The greedy algorithm used in the first stage is shown in the following algorithm.

Algorithm 1: Greedy Algorithm (First Stage)

Input: Subspace basis $\mathbf{U}$, subspace dimension r, observed vector $\mathbf{x}$, removal percentage p

Output: A reduced version $\mathbf{x}_{\boldsymbol{v}}$ of input vector $\mathbf{x}$, to be solved by MILP

1. Initialize sets of outliers as $\boldsymbol{\omega}^c = \emptyset$;
2. **for** $i = 1$ **to** d *and* $i \notin \boldsymbol{\omega}^c$ **do**
   - 2.1. Project $\mathbf{x}$ and $\mathbf{U}$ onto the existing coordinates except i and obtain $\mathbf{x}_{\boldsymbol{\omega}_i}$ and $\mathbf{U}_{\boldsymbol{\omega}_i}$;
   - 2.2. Project $\mathbf{x}_{\boldsymbol{\omega}_i}$ onto $\mathbf{U}_{\boldsymbol{\omega}_i}$ and obtain $\hat{\mathbf{x}}_{\boldsymbol{\omega}_i} = \mathbf{U}_{\boldsymbol{\omega}_i}(\mathbf{U}_{\boldsymbol{\omega}_i}^{\mathsf{T}}\mathbf{U}_{\boldsymbol{\omega}_i})^{-1}\mathbf{U}_{\boldsymbol{\omega}_i}^{\mathsf{T}}\mathbf{x}_{\boldsymbol{\omega}_i}$;
   - 2.3. Compute $r_i = \frac{\|\hat{\mathbf{x}}_{\boldsymbol{\omega}_i}\|}{\|\mathbf{x}_{\boldsymbol{\omega}_i}\|}$;
3. Find $r_m = max(r_i), \forall i \in \{1, \ldots, d\} \setminus \boldsymbol{\omega}^c$;
4. Remove m-th coordinate and update $\boldsymbol{\omega}^c = \boldsymbol{\omega}^c \cup \{m\}$;
5. Repeat steps 2 **to** 4 until we remove desired number of coordinates and thus obtain $\mathbf{x}_{\boldsymbol{v}}$;

Second Stage: MILP As output from the greedy stage, we obtain $\mathbf{x}_{\boldsymbol{v}}$—a reduced version of $\mathbf{x}$. All the entries of $\mathbf{x}_{\boldsymbol{v}}$ are not necessarily inliers. However, it is expected after being partially processed by greedy algorithm in the first stage that outliers in $\mathbf{x}_{\boldsymbol{v}}$ are less dense than in $\mathbf{x}$. We apply MILP (mixed integer linear program) formulation on $\mathbf{x}_{\boldsymbol{v}}$ which is supposed to perform better as the number of variables in $\mathbf{x}$ is reduced before feeding into MILP. The MILP for $x_{\boldsymbol{v}}$ can be formulated as follows:

$$\underset{\substack{\mathbf{z} \in \{0,1\}^{|\boldsymbol{v}|}, \\ \boldsymbol{\theta} \in \mathbb{R}^{\mathrm{r}}}}{\arg\min} \ \|\mathbf{z}\|_2^2 \quad \text{s.t.} \quad |\mathbf{x}_{\boldsymbol{v}} - \mathbf{U}_{\boldsymbol{v}}\boldsymbol{\theta}| \leq \mathrm{M}\mathbf{z}, \quad (1)$$

In this formulation, M is a sufficiently big constant useful for MILP formulation [61]. We should keep M as small as possible to avoid any potential numerical instability. Fortunately, we can use problem structure to choose M. For example, we could use

$$\mathrm{M} = \max_{\mathrm{i} \in [\mathrm{d}]} |\mathbf{U}_{\mathrm{i}}\boldsymbol{\theta} - \mathbf{x}_{\mathrm{i}}| \quad (2)$$

$\mathbf{z}$ is a vector of binary variables and assumes zero entries for inliers. To make it precise, let $\hat{\mathbf{z}}$ be the solution to (1) and $\hat{\boldsymbol{\omega}}$ be the set of indices in $\hat{\mathbf{z}}$ such that $\hat{\mathbf{z}}_{\hat{\boldsymbol{\omega}}} = \mathbf{0}$. Because $|\mathbf{x}_{\hat{\boldsymbol{\omega}}} - \mathbf{U}_{\hat{\boldsymbol{\omega}}}\boldsymbol{\theta}| \leq \mathrm{M}\hat{\mathbf{z}}_{\hat{\boldsymbol{\omega}}} = \mathbf{0}$ implies $|\mathbf{x}_{\hat{\boldsymbol{\omega}}} - \mathbf{U}_{\hat{\boldsymbol{\omega}}}\boldsymbol{\theta}| = 0$. It means all the entries indicated in $\mathbf{x}_{\hat{\boldsymbol{\omega}}}$ match with $\mathbb{U}_{\hat{\boldsymbol{\omega}}}$ for the $\boldsymbol{\theta}$ that

MILP obtains through its searching process. Moreover, M and $\mathbf{z}$ (takes 1 for outlier) collectively satisfy constraints for outliers. That is, the idea is to find a suitable $\boldsymbol{\theta}$ that makes as many inliers as possible maximizing the number of zeros (largest possible $\boldsymbol{\omega}$) in $\mathbf{z}$ or in other words, minimizing $\|\mathbf{z}\|_2^2$. Once we find correct $\boldsymbol{\theta}$ for $\mathbf{x}_\upsilon$ with at least $|\boldsymbol{\omega}| \geq r + 1$ inliers, then we can find all the inliers in $\mathbf{x}$.

MILP Formulation for Noisy Data As we mentioned earlier, high-dimensional data in different problems can be modeled as points lying in a subspace. However, in real setting, that may always not be the case. That is, there might be small modeling noise causing $\mathbf{x}_\upsilon$ to be close to $\mathbb{U}_\upsilon$ but not exactly lying on it. That is, if $\boldsymbol{\omega}$ is the set of inlier indices in $\mathbf{x}_\upsilon$, then $\mathbf{x}_\omega \approx \mathbf{U}_\omega \boldsymbol{\theta}$ holds instead of $\mathbf{x}_\omega = \mathbf{U}_\omega \boldsymbol{\theta}$ in noiseless version. Alternatively, for noise case, we can write $\mathbf{x}_\omega = \mathbf{U}_\omega \boldsymbol{\theta} + \mathbf{w}_\omega$, where $\mathbf{w}_\omega$ refers to the vector of small noises associated with inlier entries. These noises can be taken into account by a minor variation in our noiseless version (1) of MILP formulation as follows:

$$\underset{\substack{\mathbf{z} \in \{0,1\}^{|\boldsymbol{\upsilon}|}, \\ \boldsymbol{\theta} \in \mathbb{R}^r, \ \mathbf{w} \in \mathbb{R}^{|\boldsymbol{\upsilon}|}}}{\arg\min} \quad \|\mathbf{z}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \quad \text{s.t.} \quad |\mathbf{x}_\upsilon - \mathbf{U}_\upsilon \boldsymbol{\theta} - \mathbf{w}| \leq \mathbf{M}\mathbf{z}, \tag{3}$$

where $\lambda > 0$ is a regularization parameter that quantifies how much noise we will allow in our noisy formulation for the entries to be treated as inliers. $\lambda = 0$ implies that noise has no restriction on $\mathbf{w}$ in (3). Consequently, all the entries in $\mathbf{x}_\upsilon$ will be treated as inliers and we will end up with an erroneous $\boldsymbol{\theta}$. It happens because for any $\boldsymbol{\theta}$, it is possible to satisfy the constraint in (3) with $\mathbf{w} = \mathbf{x}_\upsilon - \mathbf{U}_\upsilon \boldsymbol{\theta}$ and $\mathbf{z} = \mathbf{0}$. In another extreme, if $\lambda = \infty$, it forces $\mathbf{w}$ to be $\mathbf{0}$ and allows no noise. In general, between these two extremes, if λ is smaller, the formulation in (3) will allow more noise. On the other hand, if λ is larger, it penalizes large values of $\mathbf{w}$ and thus allows less noise. For our experiments, we use moderately large value for λ. We show in Section VII that GLIMPS tolerates a reasonable amount of noise with almost no compromise with accuracy.

Remark 1. *As we already mentioned, finding* $r + 1$ *or more inliers is as good as finding* all *of the inliers in* $\mathbf{x}$*. This is because we can estimate true coefficient vector* $\boldsymbol{\theta}$ *with only a set of inliers* $\boldsymbol{\omega}$*,* $|\boldsymbol{\omega}| \geq r+1$ *using* $\boldsymbol{\theta} = (\mathbf{U}_\omega^\mathsf{T} \mathbf{U}_\omega)^{-1} \mathbf{U}_\omega^\mathsf{T} \mathbf{x}_\omega$*. As the coefficient vector* $\boldsymbol{\theta}$ *has only* r *unknowns, and we at least have* $r + 1$ *equations in the system. Consequently, the system of equations is overdetermined and the solution obtained for* $\boldsymbol{\theta}$ *is correct. Once, we obtain* $\boldsymbol{\theta}$*, we can easily obtain the entire correct* $\mathbf{x}$ *using* $\mathbf{x} = \mathbf{U}\boldsymbol{\theta}$ *even if* $\mathbf{x}$ *is partially observed or if it is severely corrupted by outliers. This idea can easily be applied to complete missing or corrupted data in applications such as background estimation when covered by foreground object or in image inpainting [59] where we want to estimate corrupted or outlier pixels in the image w.r.t the background.*

VII. Experiments

In this section, we study the performance of GLIMPS on synthetic data as a function of the fraction of outliers p, and the noise variance σ^2. For comparison, we test GLIMPS against a similar MILP formulation, the very recent greedy *Erasure* algorithm in [41] and the greedy-assisted ℓ_1-minimization (combination of greedy and ℓ_1-minimization), which tolerate larger amounts of outliers than other existing algorithms. In our experiments, we measure accuracy as the normalized error between the true and estimated inlier coefficients $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$, i.e., $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_2 / (\|\boldsymbol{\theta}\|_2 + \|\hat{\boldsymbol{\theta}}\|_2)$, and as the ratio of misclassified entries vs. total number of inliers. Both error metrics are tightly related, and all algorithms behave very similarly under both criteria.

In all our simulations, we use the ambient dimension $d = 100$ and subspace dimension $r = 5$. To represent the subspace $\mathbb{U}$ and later on, to create a sample vector $\mathbf{x}$ in the subspace, we first generate a basis $\mathbf{U} \in \mathbb{R}^{d \times r}$ and a coefficient vector $\boldsymbol{\theta} \in \mathbb{R}^r$ with i.i.d. standard gaussian entries. Then we generate $\mathbf{x}$ using $\mathbf{U}\boldsymbol{\theta} + \epsilon$, where $\epsilon \in \mathbb{R}^d$ refers to possible noise and is generated using i.i.d. $\mathcal{N}(0, \sigma^2)$. Finally, for each coordinate $i \in [d]$, with probability p, we replace the entry in $\mathbf{x}$ with an outlier generated using i.i.d standard gaussian distribution. For noisy formulation of MILP, we solve (3) with $\lambda = 1000$. We use $T = 50$ trials for all of our experiments.

We implement the greedy algorithm in Matlab and for solving MILP formulation, we use CPLEX [55] solver in AMPL environment and set a reasonable time limit (60 seconds) for each trial. CPLEX reports the best result found within the allowed amount of time. All of the experiments were run on a Macbook Pro with 3.5 GHz Intel Core i7 processor and 16 GB memory.

In our first experiment, we compare the performance of GLIMPS with other competing approaches as a function of the fraction of outliers. It is evident from Fig. 1 that GLIMPS outperforms all the other competing approaches.

As we have shown in Fig. 1 that the combined performance of greedy and ℓ_1-minimization can be achieved using only greedy algorithm that runs in $\mathcal{O}(d^2)$, we only show comparative analysis of GLIMPS with greedy and MILP formulation in the subsequent experiments.

In our second experiment, we show how the behavior of GLIMPS, the greedy *Erasure* algorithm and the MILP approach for SRMSD change with a gradual increase in fraction of outliers. The average error rate for all algorithms is shown in Fig. 2. It is observed that if we remove optimal number of outliers (in this case, 40%) in the first stage, GLIMPS can achieve 100% success rate for up to 80% outliers, whereas the greedy algorithm and MILP approach can achieve similar performance (100% success rate) for up to 60% and 76% outliers respectively.

In our third experiment, we study the computational time requirements for the greedy, MILP and proposed GLIMPS approaches. As this problem is a combinatorial problem, MILP and GLIMPS take longer time when compared with the greedy algorithm, which reduces the search space using heuristics
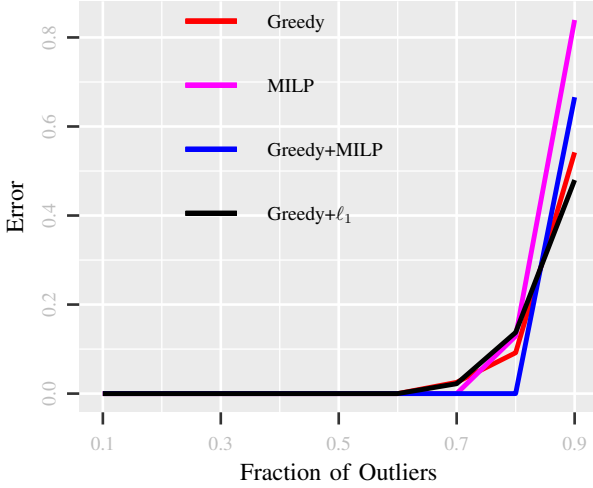
Fig. 1: GLIMPS outperforms other approaches. In the first stage of GLIMPS, we remove 40% outliers using the greedy algorithm and then in the second stage, we identify the remaining inliers and outliers using MILP. This approach succeeds for around 80% outliers. In Greedy+ℓ_1, we remove 100% outliers (empirically fixed), then use ℓ_1-minimization, but similar performance can be achieved using the greedy algorithm only.

Fig. 2: GLIMPS outperforms both the greedy and MILP approaches. In particular, GLIMPS tolerates 80% outliers whereas MILP strictly fails after 76%. Moreover, the performance of the greedy algorithm becomes unstable if the fraction of outliers goes beyond 60%.

Fig. 3: MILP and GLIMPS take longer time to solve the problem. The greedy algorithm uses heuristics to reduce the solution space and hence computationally inexpensive. In this experiment, we remove 40% in the first stage of GLIMPS.
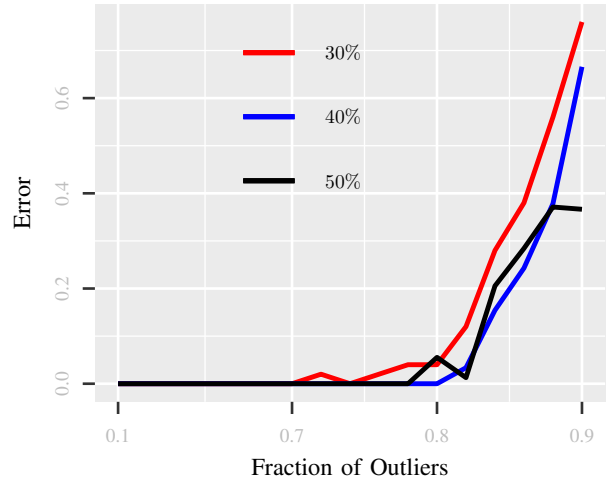

Fig. 4: We show that 40% removal in the first stage provides the best result in the second stage. Removal of 30% is low, whereas removal of 50% is ambitious in abundant-outliers regime. All three removal quantities perform very well in sparse-outliers regime.

and thus takes significantly less time (polynomial time). The results are shown in Fig. 3.

In our fourth experiment, we study the significance of outlier removal percentage in the first stage (greedy stage) of GLIMPS. We use this experiment as a step of hyperparameter tuning and apply three different removal percentages, namely 30%, 40% and 50%. The average results are shown in Fig. 4. It is noticed that 40% removal of outliers in the greedy stage achieves the best tolerance of outliers.

Finally, we study the behavior of GLIMPS under different noise conditions: *low, medium* and *high*. We use three values of sigma $\sigma = 1e^{-9}$ (low), $\sigma = 1e^{-3}$ (medium), and $\sigma = 1e^{-1}$ (high) for three levels of noises. We observe that GLIMPS steadily tolerates up to medium level of noise. However, high noise severely affects GLIMPS performance as shown in Fig. 5.

VIII. CONCLUSION

In this paper, a two-stage approach (GLIMPS) for super robust matched subspace detection is proposed to improve the
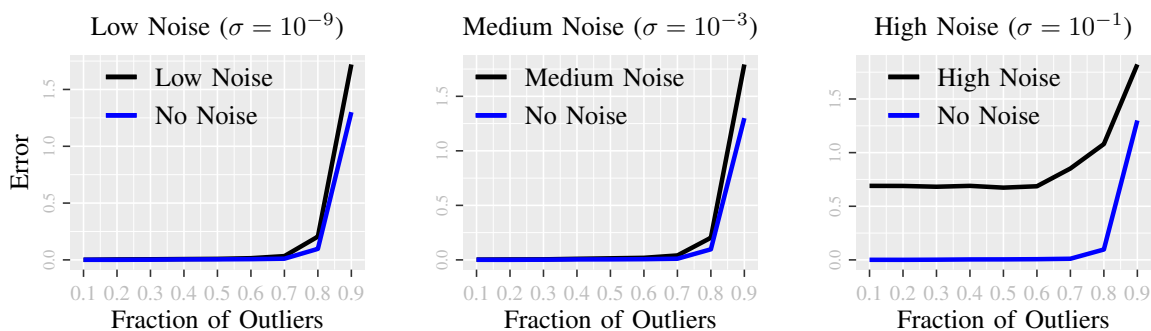
Fig. 5: GLIMPS performs satisfactorily under low and medium noise conditions. However, high noise conditions considerably affect its performance. In this experiment, 30% entries of $\mathbf{x}$ were removed in the first stage.

performance of MILP formulation in terms of both accuracy and computation time. In the first stage, a greedy algorithm is applied to remove some percentage of outliers and thus we obtain a reduced input vector. Later on, we feed the reduced input vector and warm-start MILP to increase scalability and performance of the same MILP. It is observed that GLIMPS achieves better results over *Erasure* (greedy algorithm) and MILP with full accuracy for fraction of outliers up to 80%, whereas, greedy algorithm and MILP individually can tolerate up to 60% and 76% outliers respectively. We also show that if we use greedy+ℓ_1-minimization, it can't achieve higher accuracy than only greedy algorithm. In case of GLIMPS, we notice that if outliers are abundant, greedy algorithm erroneously removes some inliers as outliers in the first stage that, in turn, restricts GLIMPS to succeed beyond 80%. Our next focus is to find necessary and sufficient deterministic theoretic conditions that will guarantee no error to be incurred in greedy algorithm (first stage) so that GLIMPS can tolerate more outliers.

REFERENCES

[1] L. Scharf and B. Friedlander, *Matched subspace detectors*, IEEE Transactions on Signal Processing, 1994.
[2] L. Scharf, *Statistical Signal Processing* Addison-Wesley, 1991.
[3] S. Shahbazpanahi, S. Valaee and M. Bastani, *Distributed source localization using ESPRIT algorithm*, IEEE Transactions on Signal Processing, 2001.
[4] M. Rangaswamy, F. Lin and K. Gerlach, *Robust adaptive signal processing methods for heterogeneous radar clutter scenarios*, Signal Processing, 2004.
[5] B. Ardekani, J. Kershaw, K. Kashikura and I. Kanno, *Activation detection in functional MRI using subspace modeling and maximum likelihood estimation*, IEEE Transactions on Medical Imaging, 1999.
[6] D. Stein, S. Beaven, L. Hoff, E. Winter, A. Schaum and A. Stocker, *Anomaly detection from hyperspectral imagery*, IEEE Signal Processing Magazine, 2002.
[7] T. Ahmed, M. Coates and A. Lakhina, *Multivariate online anomaly detection using kernel recursive least squares*, INFOCOM. 2007.
[8] M. McCloud and L. Scharf, *Interference estimation with applications to blind multiple-access communication over fading channels*, IEEE Transactions on Information Theory, 2000.
[9] K. Kanatani, *Motion segmentation by subspace separation and model selection*, IEEE International Conference in Computer Vision, 2001.
[10] H. Kwon and N. Nasrabadi, *Kernel matched subspace detectors for hyperspectral target detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006.
[11] Netflix, Inc. *The Netflix prize*, http://www.netflixprize.com.
[12] Bell, Robert M., Yehuda Koren, and Chris Volinsky. *The bellkor solution to the netflix prize.*, KorBell Team's Report to Netflix , 2007.
[13] C. Parrish, I. Jeong, R. Nowak, and R. Smith, *Empirical comparison of full-waveform lidar algorithms: Range extraction and discrimination performance*, Photogrammetric Engineering & Remote Sensing, 2011.
[14] S. Kraut, L. Schaft, L. McWhorter, *Adaptive Subspace Detectors*, IEEE Transactions on Signal Processing, 2001.
[15] M. Desai and R. Mangoubi, *Robust gaussian and non-gaussian matched subspace detection*, IEEE Transactions on Signal Processing, 2003.
[16] O. Besson, L. Scharf and F. Vincent, *Matched direction detectors and estimators for array processing with subspace steering vector uncertainties*, IEEE Transactions on Signal Processing, 2005.
[17] L. Balzano, B. Recht and R. Nowak, *High-dimensional matched subspace detection when data are missing*, IEEE International Symposium on Information Theory, 2010.
[18] L. Balzano, R. Nowak and B. Recht, *Online identification and tracking of subspaces from highly incomplete information*, Allerton Conference on Communication, Control and Computing, 2010.
[19] R. Tibshirani, *Regression Shrinkage and Selection via the Lasso*, Journal of the Royal Statistical Society, 1996.
[20] J. He, L. Balzano and A. Szlam, *Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video*, Conference on Computer Vision and Pattern Recognition, 2012.
[21] C. Papadimitriou, P. Rghavan, H. Tamaki and S. Vempala, *Latent semantic indexing, a probabilistic analysis*, Journal of Computer and System Sciences, 2000.
[22] E. Learned-Miller M. Narayana and A. Hanson, *Coherent motion segmentation in moving camera videos using optical flow orientations*, International Conference on Computer Vision, 2013.
[23] T. Bouwmans, A. Sobral, S. Javed, S. Jung and E. Zahzah, *Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset*, Computer Science Review, 2016.
[24] T. Bouwmans and E. Zahzah, *Robust PCA via principal component pursuit: a review for a comparative evaluation in video surveillance*, Computer Vision and Image Understanding, 2014.
[25] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen and Y. Ma, *Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix*, Computational Advances in Multi-Sensor Adaptive Processing, 2009.
[26] D. Pimentel-Alarcón and R. Nowak, *Random consensus robust PCA*, Electronic Journal of Statistics, 2017.
[27] H. Mansour, X. Jiang, *A robust online subspace estimation and tracking algorithm*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2015.
[28] M.A. Fischler, R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM 24 (1981) 381–395.
[29] R. Raguram, J.M. Frahm, M. Pollefeys: A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In: ECCV, Springer (2008) 500–513.
[30] R. Raguram, O. Chum, M. Pollefeys, J. Matas, J.M. Frahm, *Usac: a universal framework for random sample consensus*, IEEE TPAMI 35 (2013) 2022–2038.
[31] Y. Zheng, S. Sugimoto, and M. Okutomi, *Deterministically maximizing feasible subsystem for robust model fitting with unit norm constraint*, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 1825–1832.

[32] P. Purkait, C. Zach, and A. Eriksson, *Maximum Consensus Parameter Estimation by Reweighted ℓ_1 Methods*, International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer, Cham, 2017.

[33] T.-J. Chin, Y. Heng Kee, A. Eriksson, and F. Neumann, *Guaranteed outlier removal with mixed integer linear programs*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5858–5866.

[34] T.J. Chin and *et al.*, *Efficient globally optimal consensus maximisation with tree search*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[35] Z. Cai, and *et al.*, *Deterministic consensus maximization with biconvex programming*, arXiv preprint arXiv:1807.09436 (2018).

[36] M. C. Tsakiris and R. Vidal, *Dual principal component pursuit*, in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, pp. 10–18.

[37] H. Le, T.J. Chin, and D. Suter, *An exact penalty method for locally convergent maximum consensus*, Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE. 2017.

[38] K.H. Lee, C. Yu, and S.W. Lee, *Deterministic Hypothesis Generation for Robust Fitting of Multiple Structures*, arXiv preprint arXiv:1807.09408 (2018).

[39] R. Li, and *et al*,*ARSAC: Efficient Model Estimation via Adaptively Ranked Sample Consensus*, Neurocomputing (2018).

[40] P. Speciale and *et al.*, *Consensus Maximization with Linear Matrix Inequality Constraints*, CVPR. 2017.

[41] D. Pimentel-Alarcón, *Selective erasures for high-dimensional robust subspace tracking*, SPIE Defense + Commercial Sensing, 2018.

[42] E. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, 2009.

[43] E. Candès and T. Tao, *The power of convex relaxation: near-optimal matrix completion*, IEEE Transactions on Information Theory, 2010.

[44] B. Recht, *A simpler approach to matrix completion*, Journal of Machine Learning Research, 2011.

[45] Z. Lin, R. Liu and Z. Su, *Linearized alternating direction method with adaptive penalty for low rank representation*, Advances in Neural Information Processing Systems, 2011.

[46] P. Jain, P. Netrapalli and S. Sanghavi, *Low-rank matrix completion using alternating minimization*, ACM symposium on Theory of computing, 2013.

[47] D. Pimentel-Alarcón, N. Boston and R. Nowak, *A characterization of deterministic sampling patterns for low-rank matrix completion*, IEEE Journal of Selected Topics in Signal Processing, 2016.

[48] B. Eriksson, L. Balzano and R. Nowak, *High-rank matrix completion and subspace clustering with missing data*, Artificial Intelligence and Statistics, 2012.

[49] D. Pimentel-Alarcón, L. Balzano, R. Marcia, R. Nowak and R. Willett, *Group-sparse subspace clustering with missing data*, IEEE Statistical Signal Processing, 2016.

[50] D. Pimentel-Alarcón and R. Nowak, *The information-theoretic requirements of subspace clustering with missing data*, International Conference on Machine Learning, 2016.

[51] E. Elhamifar, *High-rank matrix completion and clustering under self-expressive models*, Advances in Neural Information Processing Systems, 2016.

[52] R. Fourer, D. Gay and B. Kernighan, *AMPL: a modeling language for mathematical programming*, Second Edition, Danvers: Boyd and Fraser.

[53] M. Lubin and I. Dunning, *Computing in Operations Research using Julia*, INFORMS Journal on Computing 27.2 (2015): 238-248.

[54] Gurobi Optimization, Inc., *Gurobi Optimizer Reference Manual*, http://www.gurobi.com, 2016.

[55] CPLEX Optimizer, https://www.ibm.com/analytics/cplex-optimizer, 2019.

[56] Albert, Réka. *Network inference, analysis, and modeling in systems biology.*, The Plant Cell 19.11 (2007): 3327-3338.

[57] B. Eriksson, P. Barford and R. Nowak, *Network Discovery from Passive Measurements*, ACM SIGCOMM, 2008.

[58] B. Eriksson, P. Barford, J. Sommers and R. Nowak, *DomainImpute: inferring unseen components in the Internet*, IEEE INFOCOM, 2011.

[59] J. Mairal, F. Bach, J. Ponce and G. Sapiro, *Online dictionary learning for sparse coding*, International Conference on Machine Learning, 2009.

[60] National Research Council, *The new science of metagenomics: revealing the secrets of our microbial planet*, National Academies Press, 2007.

[61] Chinneck, John W., *Feasibility and Infeasibility in Optimization:: Algorithms and Computational Methods.*, Vol. 118. Springer Science & Business Media, 2007.