MIXED-FEATURES VECTORS AND SUBSPACE SPLITTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Motivated by metagenomics, recommender systems, dictionary learning, and related problems, this paper introduces *subspace splitting* (SS): the task of clustering the entries of what we call a *mixed-features vector*, that is, a vector whose subsets of coordinates agree with a collection of subspaces. We derive precise identifiability conditions under which SS is well-posed, thus providing the first fundamental theory for this problem. We also propose the first three practical SS algorithms, each with advantages and disadvantages: a random sampling method , a projection-based greedy heuristic , and an alternating Lloyd-type algorithm ; all allow noise, outliers, and missing data. Our extensive experiments outline the performance of our algorithms, and in lack of other SS algorithms, for reference we compare against methods for tightly related problems, like robust matched subspace detection and maximum feasible subsystem, which are special simpler cases of SS.

1 INTRODUCTION

As the reach of data science expands, and as we continuously improve our sensing, storage and computing capabilities, data in virtually all fields of science keeps becoming increasingly high-dimensional. For example, the CERN Large Hadron Collider currently "generates so much data that scientists must discard the overwhelming majority of it, hoping that they've not thrown away anything useful" (1), and the upcoming Square Kilometer Array is expected to produce 100 times that (2). Fortunately, high-dimensional data often has an underlying low-dimensional structure. Inferring such structure not only cuts memory and computational burdens, but also reduces noise and improves learning and prediction. However, higher dimensionality not only increases computational requirements; it also augments data's structure complexity. In light of this, several research lines have explored new low-dimensional models that best summarize data, going from principal component analysis (PCA) (3; 4; 5; 6; 7; 8; 9; 10; 11) and single subspaces (12; 13; 14; 15; 16; 17; 18; 19; 20; 21) to unions of subspaces (22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40), and algebraic varieties (41).

This paper introduces *mixed-features vectors* (MFV's): a new model that describes the underlying structure of data arising from several modern applications that is not captured by existing low-dimensional models. The main idea is that each entry of a MFV comes from one out of several classes, and that the entries of the same class lie in an underlying subspace. In particular, MFV's are motivated by megatenomics (42; 43; 44; 45; 46) and recommender systems (47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59): in metagenomics each gene segment comes from one of the several taxa present in a microbiome; in recommender systems each rating may come from one of several users sharing the same account. However, MFV's also have applications in robust estimation (e.g., robust PCA (3; 4; 5; 6; 7; 8; 9; 10; 11) and robust dictionary learning (60; 61; 62; 63; 64; 65; 66; 67)), matrix completion (48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59), subspace clustering (22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 39; 32; 33; 34; 35; 36; 37; 38), and more.

This paper also introduces *subspace splitting* (SS): the task of clustering the entries of a MFV according to its underlying subspaces. SS is tightly related to other machine learning problems. In particular, SS can be thought as a generalization of robust matched subspace detection (RMSD) (12; 13; 14; 15; 16; 17), and maximum feasible subsystem (MAXFS) (68; 69; 70; 71; 72; 73; 74). However, the added complexity of SS renders existing approaches for these problems inapplicable, which calls the attention for specialized SS theory and methods. In these regards, (i) we derive

precise identifiability conditions under which SS is well-posed, and (ii) we propose the first three SS algorithms.

2 Problem Statement and Fundamental Theory

Let $\mathbb{U}^1, \dots, \mathbb{U}^{\mathrm{K}}$ be subspaces of $\mathbb{R}^{\mathrm{d}}$, and let $\Omega_0, \Omega_1, \dots, \Omega_{\mathrm{K}}$ denote a partition of $[\mathrm{d}] := \{1, \dots, \mathrm{d}\}$. For any subspace, matrix or vector that is compatible with a set of indices $\Omega \subset [\mathrm{d}]$, we will use the subscript Ω to denote its restriction to the coordinates/rows in Ω. For example, $\mathbb{U}^1_{\Omega_{\mathrm{K}}} \subset \mathbb{R}^{|\Omega_{\mathrm{K}}|}$ denotes the restriction of $\mathbb{U}^1$ to the coordinates in Ω_{K}. Define $\mathbf{x} \in \mathbb{R}^{\mathrm{d}}$ as the *mixed-features vector* (MFV) such that $\mathbf{x}_{\Omega_{\mathrm{k}}} \in \mathbb{U}^{\mathrm{k}}_{\Omega_{\mathrm{k}}}$ for each $\mathrm{k} = 1, \dots, \mathrm{K}$, and the entries of $\mathbf{x}_{\Omega_0}$ are outliers. Let $\boldsymbol{\epsilon} \in \mathbb{R}^{\mathrm{d}}$ denote a noise vector with variance σ^2. Given $\mathbb{U}^1, \dots, \mathbb{U}^{\mathrm{K}}$, and an incomplete observation $\mathbf{y}_\Omega = \mathbf{x}_\Omega + \boldsymbol{\epsilon}_\Omega$, the goal of *subspace splitting* (SS) is to determine the subsets $\Omega_1 \cap \Omega, \dots, \Omega_{\mathrm{K}} \cap \Omega$ indicating the observed coordinates of $\mathbf{y}$ that match with each subspace.

Example 1. *Consider the following setup, with* 1*-dimensional subspaces* $\mathbb{U}^1, \mathbb{U}^2$ *spanned by* $\mathbf{U}^1, \mathbf{U}^2$*:*

$$\mathbf{U}^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{U}^2 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1/2 \\ 1/2 \\ 6 \\ 8 \\ 9 \\ 10 \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} 0.1 \\ -0.1 \\ -0.1 \\ 0.1 \\ -0.1 \\ 0.1 \end{bmatrix}, \quad \mathbf{y}_\Omega = \begin{bmatrix} 0.51 \\ 0.49 \\ 5.9 \\ 8.1 \\ 8.9 \\ \cdot \end{bmatrix}.$$

It is easy to see that $\Omega_1 = \{1, 2\}$*,* $\Omega_2 = \{3, 4\}$*,* $\Omega_0 = \{5\}$*, because* $\mathbf{x}_{\Omega_1} = \frac{1}{2}\mathbf{U}^1_{\Omega_1}$ *and* $\mathbf{x}_{\Omega_2} = 2\mathbf{U}^2_{\Omega_2}$*.*

The keen reader will immediately wonder: is there another partition $\{\Omega'_1, \dots, \Omega'_{\mathrm{K}}\}$ different from $\{\Omega_1, \dots, \Omega_{\mathrm{K}}\}$ such that $\mathbf{x}_{\Omega'_{\mathrm{k}}} \in \mathbf{U}^{\mathrm{k}}_{\Omega'_{\mathrm{k}}}$ for every k? In other words, is this problem well-posed, and if so, under what conditions? Our main theoretical result answers this question, showing that under the next assumptions, Ω_{k} can be recovered if and only if it has more elements than the dimension of $\mathbb{U}^{\mathrm{k}}$.

A1 Each $\mathbb{U}^{\mathrm{k}}$ is drawn independently with respect to the uniform measure over the Grassmannian.

A2 Each $\mathbf{x}_{\Omega_{\mathrm{k}}}$ is drawn independently according to an absolutely continuous distribution with respect to the Lebesgue measure on $\mathbb{U}^{\mathrm{k}}_{\Omega_{\mathrm{k}}}$.

In words, **A1** essentially requires that $\mathbb{U}^1, \dots, \mathbb{U}^{\mathrm{K}}$ are in general position with no particular relation with one another. Similarly, **A2** requires that each *piece* of $\mathbf{x}$ is in general position over its corresponding *piece* of subspace. This type of *genericity* assumptions are becoming increasingly common in compressed sensing, matrix completion, subspace clustering, tensor theory, and related problems (10; 21; 30; 31; 41; 56; 57; 58; 33; 34; 35; 36; 37; 38; 59). All our statements hold with probability 1 with respect to the measures in **A1** and **A2**. We point out that **A1** and **A2** do not imply coherence or affinity (other typical assumptions in related theory that quantify alignment with the canonical axes or between subspaces (3; 6; 11; 23; 26; 27; 48; 49; 50; 54; 55)) nor vice-versa. For example, bounded coherence and affinity assumptions indeed allow subspaces perfectly aligned on some coordinates. However, they rule-out cases that our assumptions allow, for example the non-zero measure set of highly coherent or affine subspaces that are *somewhat* aligned with the canonical axes or with one another. To sum up, these assumptions are different, not stronger nor weaker than the usual coherence and affinity assumptions. With this, we are ready to state our main theorem, showing that subspace splitting is possible if and only if $\mathbf{x}$ contains more than $\dim(\mathbb{U}^{\mathrm{k}})$ entries of each subspace $\mathbb{U}^{\mathrm{k}}$.

Theorem 1. *Suppose **A1** and **A2** hold. Given* $\mathbf{x}$ *and* $\mathbb{U}^1, \dots, \mathbb{U}^{\mathrm{K}}$*, one can identify* $\Omega_1, \dots, \Omega_{\mathrm{K}}$ *if and only if* $|\Omega_{\mathrm{k}}| > \dim(\mathbb{U}^{\mathrm{k}})$ *for every* k.

Example 1 shows a case where the conditions of Theorem 1 are met ($|\Omega_1| = |\Omega_2| = 2 > \dim(\mathbb{U}^1) = \dim(\mathbb{U}^2) = 1$), and consequently subspace splitting is well-posed (there exist no partition other than

the true $\{\Omega_1, \Omega_2\}$ that splits $\mathbf{x}$ into $\mathbb{U}^1$ and $\mathbb{U}^2$). Conversely, the following Example shows a case where the conditions of Theorem 1 are not satisfied, and at least some Ω_k is unidentifiable.

Example 2. *Consider the following setting:*

$$\mathbf{U}^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{U}^2 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad \mathbf{U}^3 = \begin{bmatrix} 0 \\ 1 \\ 4 \\ 9 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1/2 \\ 1/2 \\ 6 \\ 3 \end{bmatrix}.$$

Now $|\Omega_2| = |\Omega_3| = 1 = \dim(\mathbb{U}^2) = \dim(\mathbb{U}^3)$. *As Theorem 1 shows, there exist multiple ways to split* $\mathbf{x}$ *into* $\mathbb{U}^1$, $\mathbb{U}^2$, *and* $\mathbb{U}^3$. *Here the partitions could be* $\{\Omega_1, \Omega_2, \Omega_3\} = \{\{1, 2\}, \{3\}, \{4\}\}$ *or* $\{\{1, 2\}, \{4\}, \{3\}\}$, *and there is no way of telling which is the* true *partition from* $\mathbb{U}^1$, $\mathbb{U}^2$, $\mathbb{U}^3$, *and* $\mathbf{x}$. *In other words,* Ω_2 *and* Ω_3 *are unidentifiable.*

Remark 1. *We point out that the constructions in our examples are not generic (i.e., they were not constructed according to* ***A1*** *and* ***A2****). We chose them for their simplicity to build intuition and make a point. However, our results still apply to these constructions, showing that in addition to* all *generic cases, our theory also holds for* some *non-generic ones. An exact characterization of* all *the non-generic cases that our theory covers requires a careful study of notions of sketching and partial coordinate discrepancy (31), which are out of the scope of this paper.*

The proof of Theorem 1 follows by the next two lemmas. Lemma 1 shows that *any* subset of r or fewer entries of *any* vector will always match with *any* r-dimensional subspace in general position. Lemma 2 shows that $r + 1$ entries of a vector won't match with a random r-dimensional subspace by chance. Said in other words, $r + 1$ entries of a vector will match with an r-dimensional subspace if and only if such entries truly come from that r-dimensional subspace. Lemma 2 is effectively the key towards Theorem 1, as it allows us to try all combinations of $r + 1$ entries that fit in an r-dimensional subspace, knowing that we will never get a false match. All proofs are in Appendix A.

Lemma 1. *Suppose* ***A1*** *holds. Let* $\Omega \subset [d]$. *If* $|\Omega| \leq \dim(\mathbb{U}^k)$, *then* $\mathbf{x}_\Omega \in \mathbb{U}^k_\Omega$ *for every* $\mathbf{x} \in \mathbb{R}^d$.

Lemma 2. *Suppose* ***A1*** *and* ***A2*** *hold. Let* Ω *be an arbitrary subset of* $[d]$ *with exactly* $\dim(\mathbb{U}^k) + 1$ *elements. Then* $\mathbf{x}_\Omega \in \mathbb{U}^k_\Omega$ *if and only if* $\Omega \subset \Omega_k$.

Corollary 1 extends these results to account for noise and missing data, replacing **A1** and **A2** with:

- **A1'** The coherence of $\mathbb{U}^k$ is upper bounded by μ, and the geodesic distance over the Grassmannian between $\mathbb{U}^k$ and $\mathbb{U}^\ell$ is lower bounded by φ, for every $k, \ell = 1, \ldots, K$.
- **A2'** The coherence of $\mathbf{x}_{\Omega_k}$ is upper bounded by ν, and its norm is lower bounded by ψ, for every $k = 1, \ldots, K$.

Corollary 1. *Suppose* ***A1'*** *and* ***A2'*** *hold with* $\mu, \nu < C\sigma$, *and* $\varphi, \psi > c\sigma$ *for some constants* C *and* c. *Let* $\mathbf{y}_\Omega$ *and* $\mathbb{U}^1, \ldots, \mathbb{U}^K$ *be given. Suppose* $|\Omega_k \cap \Omega| > \dim(\mathbb{U}^k)$ *for every* k. *Then with probability decreasing in* C *and increasing in* c, *one can identify* $\Omega_1 \cap \Omega, \ldots, \Omega_K \cap \Omega$.

To guarantee identifiability, Corollary 1 requires that subspaces and samples are sufficiently incoherent and separated to overcome the noise level σ. Notice that since now there is missing data (we observe $\mathbf{y}_\Omega$, instead of $\mathbf{x}$ as in Theorem 1), Corollary 1 requires that there are enough *observed* entries per subspace, i.e., that $|\Omega_k \cap \Omega|$ are sufficiently large (rather than Ω_k). Similarly, only the observed entires can be classified, so only the intersections $\Omega_k \cap \Omega$ are identifiable (rather than Ω_k).

3 Related Work

Arguably, the closest link that subspace splitting has with other popular machine learning problems, is through robust matched subspace detection (RMSD) (12; 13; 14; 15; 16; 17) and maximal feasible subsystem (MAXFS) (68; 69; 70; 71; 72; 73; 74). In RMSD one observes a vector $\mathbf{x} \in \mathbb{R}^d$ of the form $\mathbf{x} = \mathbf{U}\boldsymbol{\theta} + \mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^d$ is a sparse vector of *outlier* entries, and $\boldsymbol{\theta} \in \mathbb{R}^r$ is the coefficient of the entries of $\mathbf{x}$ that match with the subspace spanned by $\mathbf{U} \in \mathbb{R}^{d \times r}$. Given $\mathbf{x}$ and $\mathbf{U}$, the goal is to equivalently find $\boldsymbol{\theta}$, $\mathbf{z}$, or the support of $\mathbf{z}$, which in turn determine the coordinates Ω where $\mathbf{x}$ matches $\mathbf{U}$. Subspace splitting can be thought as the generalization of RMSD to the case where there

are multiple subspaces, and we want to find the entries that match each one. Similarly, in MAXFS one has an inconsistent system of equations of the form $\mathbf{x} = \mathbf{U}\boldsymbol{\theta}$, and wants to determine the solution $\boldsymbol{\theta}$ that produces the largest subset Ω of consistent equations. Subspace splitting can be thought as the generalization of MAXFS to the case where there are multiple consistent subsystems (one per subspace), and we want to find each of them. The three prevalent venues to solve these problems can be classified into three groups: (i) ℓ_1 minimization, (ii) mixed integer programs, and (iii) random sampling. However,none of these methodologies is applicable to subspace splitting.

The first group essentially encompasses variants of minimizing $\|\mathbf{x} - \mathbf{U}\boldsymbol{\theta}\|_1$ over $\boldsymbol{\theta} \in \mathbb{R}^{\mathrm{r}}$, which is tightly related to the Lasso (91). The intuition is that the ℓ_1-norm will favor solutions that produce a sparse vector $\mathbf{z} = \mathbf{x} - \mathbf{U}\boldsymbol{\theta}$ whose zero entries reveal Ω. While formulations like this work great for just one subspace (or subsystem), if there are two or more, then for each subspace, the entries of all other subspaces are outliers, in which case $\mathbf{z}^{\mathrm{k}} := \mathbf{x} - \mathbf{U}^{\mathrm{k}}\boldsymbol{\theta}^{\mathrm{k}}$ will no longer be sparse, and the solution to the ℓ_1-norm minimization will no longer reveal Ω_{k}. The second group aims to directly recover Ω with variants of the mixed integer formulation that minimizes $\|\mathbf{z}\|$ subject to $|\mathbf{x} - \mathbf{U}\boldsymbol{\theta}| \leq \mathrm{M}\mathbf{z}$ over $\mathbf{z} \in \{0,1\}^{\mathrm{d}}$ and $\boldsymbol{\theta} \in \mathbb{R}^{\mathrm{r}}$, where M is a tuning parameter. Here the main idea is to maximize the number of zero entries in $\mathbf{z}$ (which reveal Ω), where we are forcing $\mathbf{x}$ to match $\mathrm{span}\{\mathbf{U}\}$, because the constraint guarantees that $|\mathbf{x}_{\mathrm{i}} - \mathbf{U}\boldsymbol{\theta}| \leq \mathbf{z}_{\mathrm{i}} = 0$. Unfortunately, because of the combinatorial nature of this approach, this method becomes exponentially slow as the fraction of outliers/inconsistencies increases, rendering it prohibitive in practice for even small values of K.The third group essentially uses the same principle as random sampling consensus (RANSAC) (92; 93; 94; 95; 96; 97; 98). That is, one iteratively selects a candidate subset of entries at random, until it finds one where $\mathbf{x}$ agrees with $\mathrm{span}\{\mathbf{U}\}$. This is in fact the approach that we used in the proof of Theorem 1, and the main idea behind our first SS algorithm, which we present in Section 5.1, and study in Section 6.

Remark 2. *We point out that SS is also related to subspace clustering (22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40). The main difference is that in subspace clustering all entries of each column come from the same subspace.*

4 MOTIVATING APPLICATIONS

This section details the main motivations behind SS, namely, metagenomics, recommender systems, and robust estimation, relevant for dictionary learning, matrix completion, and related problems.

Metagenomics Microbial communities affect the balance of entire ecosystems, and play a crucial role in the planet's health (42). Consequently, understanding microbiome compositions, interactions, and evolutions holds the key to the knowledge of the deep interconnectivity factors that play a role on Earth's biodiversity, our agricultural sustainability, and adaptation to global threats like climate change. One cornerstone tool towards microbiome understanding lies in genomic analysis. In practice, to obtain an organism's genome (e.g., a person's genome), biologists feed a DNA sample (e.g., blood or hair) to a sequencer machine that produces a series of *reads*, which are short genomic sequences that can later be assembled and aligned to recover the entire genome. The challenge arises when the sequencer is provided a sample with DNA from multiple organisms, as is the case in any microbiome (e.g., the human gut microbiome), where any sample will contain a mixture of DNA from multiple taxa that cannot be trivially classified (43). In this case, each read produced by the sequencer may correspond to a different taxa, resulting in a DNA sample with a mixture of genes. Existing approaches depend on the correct classification of taxonomic units, which in turn relies on the existence of reference tables, and often require human intervention (44). However, reference sequences remain unknown for a vast majority of microbial biodiversity, which in turn precludes holistic metagenomic analyses, as scientists often have to discard unidentified data that cannot be currently categorized with existing reference tables (45). In contrast, our work pioneers the theoretical foundations of a novel model tailored to automatically classify taxonomic units without any human intervention. Moreover, our model will naturally allow for missing data, and hence it will be robust to taxa with length-varying sequences, and fast-occurring mutations, which are in fact quite common in several microorganisms such as certain types of bacteria (46).

Recommender Systems and Image Inpainting In recommender systems like Amazon or Netflix, one aims to infer users' preferences in order to make good recommendations (47). Arguably the most renowned model for this task is low-rank matrix completion (48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58), which assumes that each user's preferences vector can be explained as a linear combination of a few

others. Said in other words, preferences lie in a linear subspace. However, often multiple types of users share the same account. In this case the vector of preferences will contain a mixture of entries from multiple subspaces. It is easy to see (details in the proof of Theorem 1 in Appendix A) that the coefficient corresponding to the entries in the kth subspace are given by $\boldsymbol{\theta}^{\mathrm{k}} = (\mathbf{U}^{\mathrm{k}}_{\Omega'_{\mathrm{k}}})^{-1}\mathbf{x}_{\Omega'_{\mathrm{k}}}$, where Ω'_{k} is any subset of Ω_{k} with more than $\dim(\mathbb{U}^{\mathrm{k}})$ elements. This insight can be used to estimate the values corresponding to the kth subspace that *would* appear in the locations where there are observations from other subspaces, or where data is missing. In For example, in recommender systems, this would allow to infer the preferences of *all* users sharing an account. In image inpainting this would allow to estimate the missing, corrupted, or occluded pixels in an image (60; 61; 62; 63; 64; 65; 66; 67).

Robust Learning In Section 3 we showed that SS is a generalization of RMSD and MAXFS, both of which are crucial subtasks in many importantproblems, including robust PCA (3; 4; 5; 6; 7; 8; 9; 10; 11), dictionary learning (60; 61; 62; 63; 64; 65; 66; 67), matrix completion (48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58), and more. For example, in subspace tracking one needs to identify outliers in each new vector (meaning performing RMSD or MAXFS) before updating the underlying subspace (18; 19; 20). Or in subspace clustering one needs to identify outliers in each vector before finding sparse representations (26; 28; 29; 32; 39). These in turn are core techniques in target localization (76), medical imaging (79), communications (80), anomaly detection (81; 82), hyperspectral imaging (16), and networks estimation (**?** 84; 85; 86; 87; 88; 89), among many others (78). Being a generalization of RMSD and MAXFS, subspace splitting is also applicable in these broader problems, bringing possibilities for improved performance.Moreover, as these problems evolve into more sophisticated models that allow mixed-features vectors (such as mixture matrix completion (59)), subspace splitting will gain importance and become a crucial subroutine of these emerging problems.

5 SUBSPACE SPLITTING ALGORITHMS

We now present our three subspace splitting algorithms: a random sampling method, an iterative projection-based greedy heuristic, and an alternating approach inspired by Lloyd's algorithm (75).

5.1 RANDOM SAMPLING SPLITTING

Like Theorem 1 suggests, for each k we can iteratively try sets Ω'_{k} with $\dim(\mathbb{U}^{\mathrm{k}}) + 1$ entries selected randomly until we find one such set where $\mathbf{y}_{\Omega'_{\mathrm{k}}}$ lies close to $\mathbb{U}^{\mathrm{k}}_{\Omega'_{\mathrm{k}}}$ (within the noise level σ). At that point we can estimate $\boldsymbol{\theta}^{\mathrm{k}} = (\mathbf{U}^{\mathrm{k}}_{\Omega'})^{-1}\mathbf{y}_{\Omega'}$, and subsequently Ω_{k} as the entries in $\mathbf{y}_{\Omega} - \mathbf{U}^{\mathrm{k}}_{\Omega}\boldsymbol{\theta}^{\mathrm{k}}$ that are within σ. We call this approach *random sampling splitting* (RANSAS). The main advantage of RANSAS is that as consequence of Theorem 1, it is *guaranteed* to work after enough iterations. To see this, observe that if entries in $\mathbf{x}$ are uniformly distributed across subspaces and outliers, then the probability of randomly selecting $\mathrm{r}+1$ entries from the kth subspace is ${}^{1}\!/_{(\mathrm{K}+1)^{\mathrm{r}+1}}$. Consequently, the probability that $\mathrm{r}+1$ random entries come from the same subspace is ${}^{\mathrm{K}}\!/_{(\mathrm{K}+1)^{\mathrm{r}+1}}$. A simple Chernoff bound shows that the expected number of iterations before this happens is $\mathcal{O}(\mathrm{K}^{\mathrm{r}})$. Since we want this to happen K times, we conclude that the expected number of iterations is upper bounded by $\mathcal{O}(\mathrm{K}^{\mathrm{r}+1})$. We summarize this in the next theorem.

Theorem 2. *Let **A1** and **A2** hold. Suppose* $|\Omega_{\mathrm{k}}| > \dim(\mathbb{U}^{\mathrm{k}})$ *for every* k. *Suppose* $\mathbb{P}(\mathrm{i} \in \Omega_{\mathrm{k}}) = {}^{1}\!/_{(\mathrm{K}+1)}$ *independently for every* i, k. *Then the expected number of iterations before* RANSAS *identifies* $\Omega_1, \ldots, \Omega_{\mathrm{K}}$ *is lower and upper bounded by* $\mathcal{O}(\mathrm{K}^{\min_{\mathrm{k}} \dim(\mathbb{U}^{\mathrm{k}})+1})$ *and* $\mathcal{O}(\mathrm{K}^{\max_{\mathrm{k}} \dim(\mathbb{U}^{\mathrm{k}})+1})$.

We point out that a scenario where all entries are equally distributed is in fact the most difficult setting. Otherwise, identifying any subspace with probability $\mathrm{p_k} > {}^{1}\!/_{(\mathrm{K}+1)}$ will take $\mathcal{O}({}^{1}\!/_{\mathrm{p_k}^{\mathrm{r}+1}}) < \mathcal{O}({}^{1}\!/_{(\mathrm{K}+1)^{\mathrm{r}+1}})$ iterations to find. The main downside of this approach is that its computational complexity scales exponentially. So if K and r are small, or there is one dominant subspace, RANSAS may be a good idea. However, as we see in our experiments, this approach quickly becomes computationally prohibitive as K and r grow.

5.2 GREEDY SPLITTING

The main idea of our *greedy splitting* (GREEDYS) heuristic is to iteratively remove the most disruptive entry from each subspace. More precisely, for each k we will start with $\Omega_k^0 = [d]$, and for each $t > 0$ we will define $\Omega_k^t := \Omega_k^{t-1} \backslash i^{t-1}$, where i^{t-1} is the entry whose removal from Ω_k^{t-1} minimizes the difference between $\mathbf{y}_{\Omega_k^t}$ and its projection onto $\mathbb{U}^k_{\Omega_k^t}$. The intuition is that removing i^{t-1} gets $\mathbf{y}_{\Omega_k^t}$ closer to $\mathbb{U}^k_{\Omega_k^t}$. We repeat this procedure for each k until $\mathbf{y}_{\Omega_k^t}$ is close to $\mathbb{U}^k_{\Omega_k^t}$ (within the noise level σ). We know from Lemma 1 that this procedure will terminate at some point, because *any* subset of r or fewer entries of *any* vector will always match with *any* r-dimensional subspace in general position. If for some k this procedure terminates with a set Ω_k^t with more than $r_k = \dim(\mathbb{U}^k)$ entries, Theorem 1 suggests that $\mathbf{y}_{\Omega_k^t}$ truly lies in $\mathbb{U}^k_{\Omega_k^t}$, that $\Omega_k^t \subset \Omega$, that we can estimate $\boldsymbol{\theta}^k = (\mathbf{U}^k_{\Omega_k^t})^{-1}\mathbf{y}_{\Omega_k^t}$, and identify Ω_k as the entries in $\mathbf{y}_\Omega - \mathbf{U}^k_\Omega \boldsymbol{\theta}^k$ that are within σ. At this point we can prune all the entries corresponding to this subspace, and repeat the greedy procedure from the start without these entries. If at some point none of the remaining sets Ω_k^t has more than r_k entries, we know by Lemma 1 that we cannot determine whether $\mathbf{y}_{\Omega_k^t}$ truly lies in $\mathbb{U}^k_{\Omega_k^t}$ with these entries alone, then we stop and consider this a failure. Notice that for each k, GREEDYS removes the most disruptive entry among all d entries, then the next most disruptive among the remaining $d-1$, until either all the remaining entries lie in the same subspace, or there remain no more than r_k entries. This requires no more than $d + (d-1) + (d-2) + \cdots + (r_k + 1) = \mathcal{O}(d^2)$ iterations. After identifying the entries of a subspace, if pruning is necessary, we potentially need to repeat the iterative removal process for the remaining $K-1$ subspaces, and so on. In total, we potentially need to repeat the iterative removal process $K + (K-1) + (K-2) + \cdots + 1 = \mathcal{O}(K^2)$ times. Putting these two observations together, we obtain the following theorem, showing that GREEDYS has a polynomial computational complexity.

Theorem 3. GREEDYS *will terminate after no more than* $\mathcal{O}(Kd)^2$ *iterations.*

The main caveat of this greedy approach, as the next Example shows, is that even in a noiseless setting, the most disruptive entry for $\mathbb{U}^k$ (that is, the entry whose removal minimizes the distance between $\mathbf{y}_{\Omega_k^t}$ and its projection onto $\mathbb{U}^k_{\Omega_k^t}$) may in fact correspond to $\mathbb{U}^k$.

Example 3. *Consider the following construction where the first two entries of* $\mathbf{x}$ *correspond to* $\mathbf{U}^1$*, the last two correspond to* $\mathbf{U}^2$*, there are no outliers, i.e.,* $\Omega_0 = \emptyset$*, there is no noise, i.e.,* $\boldsymbol{\epsilon} = \mathbf{0}$*, and there is no missing data, i.e.,* $\Omega = [d]$*, so that we observe all entries of* $\mathbf{y} = \mathbf{x}$*:*

$$\mathbf{U}^1 = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \qquad \mathbf{U}^2 = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 1 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 4 \\ 4 \\ 2 \\ 2 \end{bmatrix}.$$

Pay special attention to $\mathbf{U}^1$ *and* $\mathbf{y}$*. Let* $\Omega_{-i} := [d] \backslash i$ *denote the set of all except the* i^{th} *entry, and let* $\hat{\mathbf{y}}^k_{\Omega_{-i}} := \mathbf{U}^k_{\Omega_{-i}}(\mathbf{U}^{k\mathsf{T}}_{\Omega_{-i}}\mathbf{U}^k_{\Omega_{-i}})^{-1}\mathbf{U}^{k\mathsf{T}}_{\Omega_{-i}}\hat{\mathbf{y}}_{\Omega_{-i}}$ *denote the projection of* $\mathbf{y}_{\Omega_{-i}}$ *onto* $\mathbb{U}^k_{\Omega_{-i}}$*. For example, here:*

$$\mathbf{U}^1_{\Omega_{-1}} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \qquad \mathbf{y}_{\Omega_{-1}} = \begin{bmatrix} 4 \\ 2 \\ 2 \end{bmatrix}, \qquad \hat{\mathbf{y}}^1_{\Omega_{-1}} = \begin{bmatrix} 1.572 \\ 2.2759 \\ 3.0345 \end{bmatrix}.$$

Proceeding according to our greedy heuristic, we can compute the projection residuals when removing each entry:

$$\|\mathbf{y}_{\Omega_{-1}} - \hat{\mathbf{x}}^1_{\Omega_{-1}}\| = 2.7038, \qquad \|\mathbf{y}_{\Omega_{-2}} - \hat{\mathbf{x}}^1_{\Omega_{-2}}\| = 2.7038,$$
$$\|\mathbf{y}_{\Omega_{-3}} - \hat{\mathbf{x}}^1_{\Omega_{-3}}\| = 3.4641, \qquad \|\mathbf{y}_{\Omega_{-4}} - \hat{\mathbf{x}}^1_{\Omega_{-4}}\| = 2.7440.$$

Here we see that removing any of the first two entries makes $\mathbf{y}$ appear *closer to* $\mathbb{U}^1$*, even though the first two entries in fact correspond to* $\mathbb{U}^1$*. We thus conclude that our greedy heuristic, while it works well in practice, it has no theoretical guarantee of working.*

5.3 K-SPLITS

As the name suggests, our last algorithm, K-SPLITS, is inspired by the K-means algorithm. The main idea is to initialize partitioning randomly the entries of $\mathbf{y}_\Omega$ into $\hat{\Omega}_0, \hat{\Omega}_1, \ldots, \hat{\Omega}_K$, and then alternate

until convergence between (i) estimating coefficients $\hat{\boldsymbol{\theta}}^1, \ldots, \hat{\boldsymbol{\theta}}^{\mathrm{K}}$ in light of the assignment of entries, and (ii) reassigning entries according to these coefficients. More precisely, at each step we:

(i) Estimate coefficients $\hat{\boldsymbol{\theta}}^{\mathrm{k}} = (\mathbf{U}^{\mathrm{k}}_{\hat{\Omega}_{\mathrm{k}}})^{-1}\mathbf{y}_{\hat{\Omega}_{\mathrm{k}}}$.

(ii) Compute $\hat{\mathbf{y}}^{\mathrm{k}} := \mathbf{U}^{\mathrm{k}}\hat{\boldsymbol{\theta}}^{\mathrm{k}}$, and assign entries as $\hat{\Omega}_{\mathrm{k}} = \left\{\mathrm{i} \in [\mathrm{d}] \ : \ |\mathbf{y}_{\mathrm{i}} - \hat{\mathbf{y}}^{\mathrm{k}}_{\mathrm{i}}| \leq \tau(\sigma)\right\}$, where $\tau(\sigma)$ is a tuning thresholding parameter that depends on the noise level.

As we will see in our experiments, this method outperforms the rest. Its main downside, however, is that like most alternating methods, it suffers from local minima, and depends heavily on initialization. To mitigate the former, in our experiments we run several independent random initializations. We observe that in practice this has excellent performance. We conjecture that it is because this strategy will at some point spread the *centers* $\boldsymbol{\theta}^{\mathrm{k}}$ enough, in light of the subspaces. This sort of spread is crucial in other related clustering algorithms, such as K-means (75). Our future work will further investigate this, convergence, complexity, and initialization strategies, such as parallels of K-means++ (99), which require careful analysis and are out of the scope of this paper.

6 EXPERIMENTS

In this section we study the performance of our three methods above, as a function of the number of subspaces K, the number of entries per subspace $|\Omega_{\mathrm{k}}|$ (which in turn is a proxy of the ambient dimension d) and the subspaces dimensions r. We measure error as the fraction of misclassified entries. In lack of other existing subspace splitting methods, for reference we compare them against the ℓ_1 minimization described in Section 3. Our results show that our SS methods perform as well as existing methods in the single subspace case, and succeed in the presence of multiple subspaces, where existing methods fail dramatically.

In our experiments we first generate matrices $\mathbf{U}^1, \ldots, \mathbf{U}^{\mathrm{K}} \in \mathbb{R}^{\mathrm{d}\times\mathrm{r}}$ with i.i.d. $\mathcal{N}(0,1)$ entries, to use as bases of $\mathbb{U}^1, \ldots, \mathbb{U}^{\mathrm{K}}$, with $\mathrm{d} = (\mathrm{K}+1)|\Omega_{\mathrm{k}}|$. Similarly, we generate $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{\mathrm{K}} \in \mathbb{R}^{\mathrm{r}}$, also with i.i.d. $\mathcal{N}(0,1)$ entries, to use as coefficient vectors. Next we create the partition of entries $\{\Omega_0, \Omega_1, \ldots, \Omega_{\mathrm{K}}\}$ where each Ω_{k} has the exact same number of elements. As discussed in Section 5.1, this is in fact the most difficult setting; otherwise the dominant subspace is easier to find, and one can simply prune its corresponding entries to simplify the problem. Then we create $\mathbf{x}$ as the vector such that $\mathbf{x}_{\Omega_{\mathrm{k}}} = \mathbf{U}^{\mathrm{k}}_{\Omega_{\mathrm{k}}}\boldsymbol{\theta}^{\mathrm{k}}$ for each k, and $\mathbf{x}_{\Omega_0}$ are i.i.d. $\mathcal{N}(0,1)$ entries representing outliers. Finally, we generate $\boldsymbol{\epsilon} \in \mathbb{R}^{\mathrm{d}}$ with i.i.d. $\mathcal{N}(0,\sigma^2)$ entries and $\Omega \subset \mathrm{d}$ with a fraction p of observed entries, to construct our observed data $\mathbf{y}_{\Omega} = \mathbf{x}_{\Omega} + \boldsymbol{\epsilon}_{\Omega}$.

In our first experiment we study the behavior of our algorithms as a function of K, with $\mathrm{r} = 5$, $|\Omega_{\mathrm{k}}| = 40$, $\mathrm{p} = 1/2$ (so that on average 20 observed entries correspond to each subspace), and $\sigma = 0$, i.e., no noise. Figure 1 (a) shows that even for $\mathrm{K} = 2$ (the smallest value of K), existing methodology is unable to split $\mathbf{y}$. This is not surprising, as existing methods are meant for settings where *most* of the entries in $\mathbf{y}$ agree with a single subspace, and there are only a few outliers. In our setting, for each subspace, the entries of all other subspaces are outliers. Consequently, a fraction $1 - 1/K$ of the entries (at least half with $\mathrm{K} = 2$, and a dominant majority as K increases) are outliers for each subspace, which is well-documented to make existing methods fail (see the discussion in Section 3 for details). In contrast, GREEDYS is better with $\mathrm{K} = 6$ (which amounts to 83.3% outliers) than ℓ_1 is with $\mathrm{K} = 2$, while K-SPLITS and RANSAS maintain a 100% success rate for every K we tried. Nonetheless, increasing in K comes at a price (time), as seen in Figure 1 (a'), which shows how the number of iterations of each algorithm grows with K, relative to the simplest case ($\mathrm{K} = 2$). Notice that the speed of K-SPLITS decays nicely with K. For instance, with $\mathrm{K} = 6$, K-SPLITS degrades an order of magnitude slower than RANSAS, while still maintaining the same 100% success rate.

In our second experiment we study the behavior of our algorithms as a function of $|\Omega_{\mathrm{k}}|$, with $\mathrm{r} = 5$, $\mathrm{p} = 1/2$, $\sigma = 0$, and $\mathrm{K} = 2$ fixed. Figures 1 (b) and (b') show that as $|\Omega_{\mathrm{k}}|$ grows, the performance of all methods improves: ℓ_1 and GREEDYS achieve higher success rates, while RANSAS and K-SPLITS increase their speed and maintain the same 100% success rate.

In our third experiment we study performance as a function of r, with $\mathrm{p} = 1/2$, $\sigma = 0$, and $\mathrm{K} = 2$ fixed. Recall from Lemma 1 that as r increases, so does the number of entries per subspace required for identifiability $(\mathrm{r}+1)$. Hence to allow a wide range of r, we fixed $|\Omega_{\mathrm{k}}| = 40$. Figures 1 (c) and

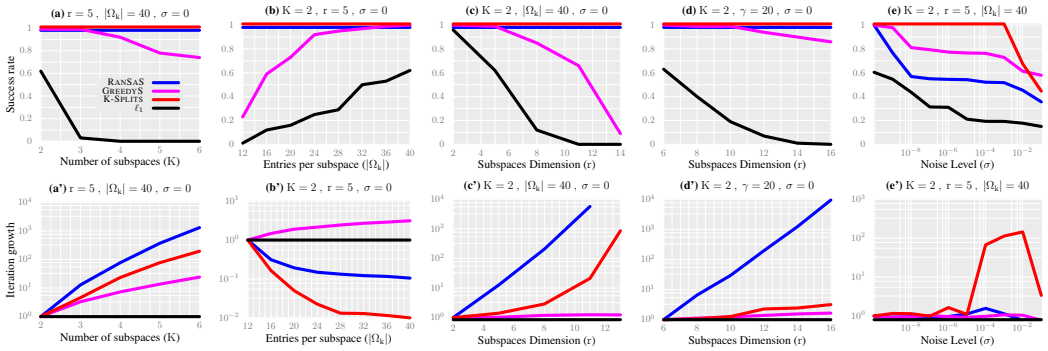

Figure 1: Average success rate and iteration growth over 100 trials, as a function of the number of subspaces K, the number of entries per subspace $|\Omega_{\mathrm{k}}|$ (which in turn is a proxy of the ambient dimension d), the subspaces dimensions r, the gap $\gamma := |\Omega_{\mathrm{k}}| - (\mathrm{r}+1)$, and the noise level σ.

(c'), show that (conversely to our previous experiment) the performance of all methods declines as r grows: ℓ_1 and GREEDYS achieve lower success rates, while RANSAS and K-SPLITS increase their time (though they maintain the same 100% success rate).

Our next two experiments are consistent with the intuition that cases with fewer data (small $|\Omega_{\mathrm{k}}|$) relative to the problem's complexity (the dimension r) are harder. We verify this in our next experiments, where we study performance as a function of r, fixing $\mathrm{p} = 1/2$, $\sigma = 0$, and $\mathrm{K} = 2$ and the gap $\gamma := \mathrm{p}(|\Omega_{\mathrm{k}}| - (\mathrm{r}+1)) = 20$. This setting represents the case where we always have a few more (20) entries per subspace than the strictly required $(\mathrm{r}+1)$ for splitting. This is arguably a better experiment to test the effect of the subspaces dimensions isolated from the sampling rate. Figures 1 (d) and (d') now show that r slightly affects GREEDYS and K-SPLITS(GREEDYS in terms of accuracy, and K-SPLITS in terms of time), but dramatically affects RANSAS, causing an increase of iterations four orders of magnitude higher than the rest. This is consistent with our discussion in Section 5.1, showing that RANSAC will always succeed given enough time, which grows exponentially in r. Interestingly, K-SPLITS achieves the same 100% performance, but orders of magnitude faster, suggesting that K-SPLITS is in general the most promising SS algorithm.

In our last experiment we study the accuracy of our algorithms as a function of the noise level σ, with $\mathrm{r} = 5$, $\mathrm{p} = 1/2$, $|\Omega_{\mathrm{k}}| = 40$, and $\mathrm{K} = 2$ fixed. Consistent with Corollary 1, Figure 1 (e) shows that the performance of our algorithms decays nicely with noise, with K-SPLITS allowing up to $\sigma = 10^{-3}$. The price, however, can be seen in execution time (Figure 1 (e')), which grows up to the point where algorithms start to decay, in which case their execution time also starts decreasing. We point out that the fraction of missing data and outliers only affect results and performance in the sense that they reduce the number of available observed entries per subspace, so their effects are also captured in the experiments on $|\Omega_{\mathrm{k}}|$. For example, a case where there are $|\Omega_{\mathrm{k}}| = 40$ samples per subspace, but only half of the entries are observed ($\mathrm{p} = 1/2$) is equivalent to observing $|\Omega_{\mathrm{k}}| = 20$ samples per subspace with no missing data ($\mathrm{p} = 1$). Appendix B shows additional results with $\mathrm{K} = 4$, where our methods show even more dramatic improvements.

BROADER IMPACT

This paper introduces a new dimensionality reduction model, and a series of methods to infer such model, with broad applicability ranging from metagenomics to recommender systems. From a pragmatic standpoint, practitioners can use this research to better predict drug-target interactions, analyze the composition and dynamics soil or human microbiomes, and more. Ultimately, this can impact the well-being of society at many levels, improving drugs, medical diagnoses, treatments, agricultural sustainability, etc. From a learning perspective, we hope this work spurs discussions and insights, and motivates the data science community to explore new directions that stem from this initial work (for example near-optimal initialization strategies on our K-SPLITS algorithm, similar to K-means++), and that leads to better methods and understanding of subspace splitting.

REFERENCES

[1] Edd Wilder-James, *What is big data? An introduction to the big data landscape*, available at: https://www.oreilly.com/ideas/what-is-big-data, 2018

[2] *SKA Signs Big Data Cooperation Agreement With CERN*, available at: https://www.skatelescope.org/news/ska-signs-big-data-cooperation-agreement-cern/, 2018.

[3] X. Yuan and J. Yang, *Sparse and low-rank matrix decomposition via alternating direction methods*, available at http://www.optimization-online.org/DB_HTML/2009/11/2447.html, 2009.

[4] 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.

[5] X. Ding, L. He and L. Carin, *Bayesian robust principal component analysis*, IEEE Transactions on Image Processing, 2011.

[6] E. Candès, X. Li, Y. Ma and J. Wright, *Robust principal component analysis?*, Journal of the ACM, 2011.

[7] 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.

[8] Y. Ma, *Low-rank matrix recovery and completion via convex optimization*, avaiable at http://perception.csl.illinois.edu/matrix-rank/home.html.

[9] 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.

[10] D. Pimentel-Alarcón and R. Nowak, *Random consensus robust PCA*, Electronic Journal of Statistics, 2017.

[11] M. C. Tsakiris and R. Vidal, *Dual principal component pursuit*, Journal of Machine Learning Research, 2018.

[12] L. Scharf and B. Friedlander, *Matched subspace detectors*, IEEE Transactions on Signal Processing, 1994.

[13] S. Kraut, L. Schaft, L. McWhorter, *Adaptive Subspace Detectors*, IEEE Transactions on Signal Processing, 2001.

[14] M. Desai and R. Mangoubi, *Robust gaussian and non-gaussian matched subspace detection*, IEEE Transactions on Signal Processing, 2003.

[15] 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.

[16] H. Kwon and N. Nasrabadi, *Kernel matched subspace detectors for hyperspectral target detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006.

[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] 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.

[20] H. Mansour, X. Jiang, *A robust online subspace estimation and tracking algorithm*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2015.

[21] D. Pimentel-Alarcón, N. Boston and R. Nowak, *Deterministic conditions for subspace identifiability from incomplete sampling*, IEEE International Symposium on Information Theory, 2015.

[22] K. Kanatani, *Motion segmentation by subspace separation and model selection*, IEEE International Conference in Computer Vision, 2001.

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

[24] L. Balzano, A. Szlam, B. Recht, and R. Nowak, *K-subspaces with missing data*, IEEE Statistical Signal Processing, 2012.

[25] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, *Robust recovery of subspace structures by low-rank representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012.

[26] E. Elhamifar and R. Vidal, *Sparse subspace clustering: algorithm, theory, and applications*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013.

[27] D. Pimentel-Alarcón, L. Balzano and R. Nowak, *On the sample complexity of subspace clustering with missing data*, IEEE Statistical Signal Processing, 2014.

[28] C. Yang, D. Robinson and R. Vidal, *Sparse subspace clustering with missing entries*, International Conference on Machine Learning, 2015.

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

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

[31] D. Pimentel-Alarcón, L. Balzano and R. NowakNecessary and sufficient conditions for sketched subspace clustering, Allerton Conference on Communication, Control and Computing, 2016.

[32] Y. Chong, D. Robinson and R. Vidal, *Provable self representation based outlier detection in a union of subspaces*, IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[33] M. Ashraphijuo, X. Wang and V. Aggarwal, *A characterization of sampling patterns for low-rank multi-view data completion problem*, IEEE International Symposium on Information Theory, 2017.

[34] M. Ashraphijuo, V. Aggarwal and X. Wang, *A characterization of sampling patterns for low-tucker-rank tensor completion problem*, IEEE International Symposium on Information Theory, 2017.

[35] M. Ashraphijuo and X. Wang, *Fundamental conditions for low-CP-rank tensor completion*, Journal of Machine Learning Research, 2017.

[36] M. Ashraphijuo, V. Aggarwal and X. Wang, *Deterministic and probabilistic conditions for finite completability of low-tucker-rank tensor*, IEEE Transactions on Information Theory, 2019.

[37] M. Ashraphijuo and X. Wang, *Clustering a union of low-rank subspaces of different dimensions with missing data*, Pattern Recognition Letters 2019.

[38] M. Ashraphijuo, X. Wang, *Fundamental conditions on the sampling pattern for union of low-rank subspaces retrieval*, Annals of Mathematics and Artificial Intelligence, 2019.

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

[40] G. Ongie, R. Willett, R. Nowak and L. Balzano, *Algebraic variety models for high-rank matrix completion*, International Conference on Machine Learning, 2017.

[41] G. Ongie, D. Pimentel, L. Balzano, R. Willett and R. Nowak, *Low algebraic dimension matrix completion*, Allerton Conference on Communication, Control and Computing, 2017.

[42] S. Highlander, *High throughput sequencing methods for microbiome profiling: application to food animal systems*, Animal Health Research Reviews, 2012.

[43] S. Mande, M. Mohammed and T. Ghosh, *Classification of metagenomic sequences: methods and challenges*, Briefings in Bioinformatics, 2012.

[44] R. Ranjan, A. Rani, A. Metwally, H. McGee and D. Perkins, *Analysis of the microbiome: Advantages of whole genome shotgun versus 16S amplicon sequencing*, Biochemical and Biophysical Research Communications, 2016.

[45] N. Nguyen, T. Warnow, M. Pop and B. White, *A perspective on 16S rRNA operational taxonomic unit clustering using sequence similarity*, Biofilms and Microbiomes, 2016.

[46] G. Marçais, A. Delcher, A. Phillippy, R. Coston, S. Salzberg and A. Zimin, *MUMmer4: A fast and versatile genome alignment system*, PLoS Computational Biology, 2018.

[47] Netflix, Inc. *The Netflix prize*, `http://www.netflixprize.com`.

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

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

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

[51] 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.

[52] Z. Wen, W. Yin and Y. Zhang, *Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm*, Mathematical Programming Computation, 2012.

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

[54] Y. Chen, S. Bhojanapalli, S. Sanghavi and R. Ward, *Coherent matrix completion*, International Conference on Machine Learning, 2014.

[55] Y. Chen, *Incoherent-optimal matrix completion*, IEEE Transactions on Information Theory, 2015.

[56] F. Király, L. Theran and R. Tomioka, *The algebraic combinatorial approach for low-rank matrix completion*, Journal of Machine Learning Research, 2015.

[57] 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.

[58] D. Pimentel-Alarcón and R. Nowak, *A converse to low-rank matrix completion*, IEEE International Symposium on Information Theory, 2016.

[59] D. Pimentel-Alarcón, *Mixture Matrix Completion*, Advances in Neural Information Processing Systems, 2018.

[60] M. Aharon, M. Elad and A. Bruckstein, *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*, IEEE Transactions on Signal Processing, 2006.

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

[62] C. Zhao, X. Wang and W. Cham, *Background subtraction via robust dictionary learning*, EURASIP Journal on Image and Video Processing, 2011

[63] C. Lu, J. Shi and J. Jia, *Online robust dictionary learning*, IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[64] A. Iqbal and A. Seghouane, *An α-divergence-based approach for robust dictionary learning*, IEEE Transactions on Image Processing, 2019.

[65] Y. Xu, Z. Li, C. Tian and J. Yang, *Multiple vector representations of images and robust dictionary learning*, Pattern Recognition Letters, 2019.

[66] S. Mahdizadehaghdam, A. Panahi, H. Krim and L. Dai, *Deep dictionary learning: A parametric network approach*, IEEE Transactions on Image Processing, 2019.

[67] J. Ren, Z. Zhang, S. Li, Y. Wang, G. Liu, S. Yan, and M. Wang, *Learning Hybrid Representation by Robust Dictionary Learning in Factorized Compressed Space* IEEE Transactions on Image Processing, 2020.

[68] Y. Zheng, S. Sugimoto, and M. Okutomi, *Deterministically maximizing feasible subsystem for robust model fitting with unit norm constraint*, IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[69] 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.

[70] T.J. Chin, P. Purkait, A. Eriksson and D. Suter, *Efficient globally optimal consensus maximisation with tree search*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

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

[72] P. Speciale, D. Pani, M. Oswald, T. Kroeger, L. Van Gool and M. Pollefeys, *Consensus Maximization with Linear Matrix Inequality Constraints*, IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[73] Z. Cai, T, Chin, H. Le and D. Suter, *Deterministic consensus maximization with biconvex programming*, arXiv preprint, 2018.

[74] C. Yu and D.Y. Ju, *A maximum feasible subsystem for globally optimal 3D point cloud registration*, Sensors, 2018.

[75] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Wu, *An efficient k-means clustering algorithm: Analysis and implementation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002.

[76] S. Shahbazpanahi, S. Valaee and M. Bastani, *Distributed source localization using ESPRIT algorithm*, IEEE Transactions on Signal Processing, 2001.

[77] M. Rangaswamy, F. Lin and K. Gerlach, *Robust adaptive signal processing methods for heterogeneous radar clutter scenarios*, Signal Processing, 2004.

[78] H. Krim and M. Viberg, *Two decades of array signal processing research: the parametric approach*, Signal Processing Magazine, 1996.

[79] 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.

[80] M. McCloud and L. Scharf, *Interference estimation with applications to blind multiple-access communication over fading channels*, IEEE Transactions on Information Theory, 2000.

[81] D. Stein, S. Beaven, L. Hoff, E. Winter, A. Schaum and A. Stocker, *Anomaly detection from hyperspectral imagery*, IEEE Signal Processing Magazine, 2002.

[82] T. Ahmed, M. Coates and A. Lakhina, *Multivariate online anomaly detection using kernel recursive least squares*, INFOCOM. 2007.

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

[84] R. Govindan and H. Tangmunarunkit, *Heuristics for Internet Map Discovery*, IEEE INFOCOM 2000.

[85] P. Barford, A. Bestavros, J. Byers and M. Crovella, *On the marginal utility of network topology measurements*, Proceedings of ACM Internet Measurement Workshop, 2001.

[86] N. Spring, R. Mahajan, D. Wetherall and T. Anderson, *Measuring ISP topologies with rocketfuel*, IEEE/ACM Transactions on Networking, 2004.

[87] D. Alderson, L. Li, W. Willinger and J. Doyle, *Understanding internet topology: Principles, models and validation*, IEEE/ACM Transactions on Networking, 2005.

[88] R. Sherwood, A. Bender and N. Spring, *DisCarte: A disjunctive Internet cartographer*, ACM SIGCOMM, 2008.

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

[90] 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.

[91] R. Tibshirani, *Regression Shrinkage and Selection via the Lasso*, Journal of the Royal Statistical Society, 1996.

[92] 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, 1981.

[93] R. Raguram, J.M. Frahm, M. Pollefeys: *A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus*, European Conference in Computer Vision, 2008.

[94] R. Raguram, O. Chum, M. Pollefeys, J. Matas, J.M. Frahm, *Usac: a universal framework for random sample consensus*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013.

[95] M. Feigin, B.J. Ranger, and B.W. Anthony, *Statistical consensus matching framework for image registration*, International Conference on Pattern Recognition, 2016.

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

[97] R. Li, J. Sun, D. Gong, Y. Zhu, H. Li and Y. Zhang, *ARSAC: Efficient Model Estimation via Adaptively Ranked Sample Consensus*, Neurocomputing, 2018.

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

[99] D. Arthur and S. Vassilvitskii, *k-means++: The advantages of careful seeding*, Annual ACM-SIAM Symposium on Discrete Algorithms, 2007.

A PROOFS

In this section we present the proofs of all statements. We begin with the proof of Lemmas 1 and 2, which are required for the proof of Theorem 1.

Proof. (Lemma 1) Since $\mathbb{U}^{\mathrm{k}}$ is in general position, if $|\Omega| \leq \dim(\mathbb{U}^{\mathrm{k}})$, then $\mathbb{U}^{\mathrm{k}}_{\Omega} = \mathbb{R}^{|\Omega|}$. In other words, $\mathbb{U}^{\mathrm{k}}_{\Omega}$ is the entire $|\Omega|$-dimensional space. □

Proof. (Lemma 2) If $\Omega \subset \Omega_{\mathrm{k}}$ then $\mathbf{x}_{\Omega}$ trivially lies in $\mathbb{U}^{\mathrm{k}}_{\Omega}$ by **A2**. To see the reverse implication, let $\mathrm{r} := \dim(\mathbb{U}^{\mathrm{k}})$. We know that $\mathbf{x}_{\Omega} \in \mathbb{U}^{\mathrm{k}}_{\Omega}$ if and only if there exists a coefficient vector $\boldsymbol{\theta} \in \mathbb{R}^{\mathrm{r}}$ such that

$$\mathbf{x}_{\Omega} \;=\; \mathbf{U}^{\mathrm{k}}_{\Omega}\boldsymbol{\theta}, \tag{1}$$

where $\mathbf{U}^{\mathrm{k}} \in \mathbb{R}^{\mathrm{d}\times\mathrm{r}}$ denotes a basis of $\mathbb{U}^{\mathrm{k}}$. Since $|\Omega| = \mathrm{r}+1$ by assumption, equation 1 defines a system of $\mathrm{r}+1$ equations (the rows of $\mathbf{x}_{\Omega}$) and r unknowns (the entries of $\boldsymbol{\theta}$). Assume without loss of generality that $\Omega = \{1, 2, \ldots, \mathrm{r}+1\}$ (otherwise simply permute the rows accordingly), and let $\Omega' = \{1, 2, \ldots, \mathrm{r}\}$ be the subset of Ω with the first r elements. Then we can rewrite equation 1 as

$$\begin{array}{c} \mathrm{r} \\ 1 \end{array} \left\{ \begin{array}{c} \\ \\ \end{array} \right. \left[\begin{array}{c} \mathbf{x}_{\Omega'} \\ \hline \mathbf{x}_{\mathrm{r}+1} \end{array}\right] = \left[\begin{array}{c} \mathbf{U}^{\mathrm{k}}_{\Omega'} \\ \hline \mathbf{U}^{\mathrm{k}}_{\mathrm{r}+1} \end{array}\right] \boldsymbol{\theta}.$$

Notice that $\mathbf{U}^{\mathrm{k}}_{\Omega'} \in \mathbb{R}^{\mathrm{r}\times\mathrm{r}}$ is invertible because **A1** guarantees that $\mathbb{U}^{\mathrm{k}}$ is in general position with probability 1. Hence we can use the top block to obtain $\boldsymbol{\theta} = (\mathbf{U}^{\mathrm{k}}_{\Omega'})^{-1}\mathbf{x}_{\Omega'}$, and we can plug this in the last row to obtain:

$$\mathbf{x}_{\mathrm{r}+1} \;=\; \mathbf{U}^{\mathrm{k}}_{\mathrm{r}+1}(\mathbf{U}^{\mathrm{k}}_{\Omega'})^{-1}\mathbf{x}_{\Omega'}.$$

Let $\mathrm{k_i}$ indicate the subspace corresponding to the ith entry of $\mathbf{x}$. Then by **A2** there exist coefficients $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{\mathrm{K}}$ such that $\mathbf{x}_{\mathrm{i}} = \mathbf{U}^{\mathrm{k_i}}_{\mathrm{i}}\boldsymbol{\theta}^{\mathrm{k_i}}$. In light of this we can rewrite the last equation as

$$\mathbf{U}^{\mathrm{k_{r+1}}}_{\mathrm{r}+1}\boldsymbol{\theta}^{\mathrm{k_{r+1}}} \;=\; \mathbf{U}^{\mathrm{k}}_{\mathrm{r}+1}(\mathbf{U}^{\mathrm{k}}_{\Omega'})^{-1}\begin{bmatrix} \mathbf{U}^{\mathrm{k_1}}_{1}\boldsymbol{\theta}^{\mathrm{k_1}} \\ \mathbf{U}^{\mathrm{k_2}}_{2}\boldsymbol{\theta}^{\mathrm{k_2}} \\ \vdots \\ \mathbf{U}^{\mathrm{k_r}}_{\mathrm{r}}\boldsymbol{\theta}^{\mathrm{k_r}} \end{bmatrix}. \tag{2}$$

At this point equation 2 has no variables left (all $\mathbf{U}^{\mathrm{k}}$'s and $\boldsymbol{\theta}^{\mathrm{k}}$'s are fixed), which means that equation 2 is a consistency condition. Notice that if $\Omega \subset \Omega_{\mathrm{k}}$, then $\mathrm{k_i} = \mathrm{k}$, and equation 2 transforms into

$$\mathbf{U}^{\mathrm{k}}_{\mathrm{r}+1}\boldsymbol{\theta}^{\mathrm{k}} = \mathbf{U}^{\mathrm{k}}_{\mathrm{r}+1}(\mathbf{U}^{\mathrm{k}}_{\Omega'})^{-1}\mathbf{U}^{\mathrm{k}}_{\Omega'}\boldsymbol{\theta}^{\mathrm{k}},$$

which further simplifies to $\mathbf{U}^{\mathrm{k}}_{\mathrm{r}+1}\boldsymbol{\theta}^{\mathrm{k}} = \mathbf{U}^{\mathrm{k}}_{\mathrm{r}+1}\boldsymbol{\theta}^{\mathrm{k}}$. In other words, if $\Omega \subset \Omega_{\mathrm{k}}$, the consistency condition equation 2 is met. On the other hand, if $\Omega \not\subset \Omega_{\mathrm{k}}$, equation 2 does not simplify any further. Recall from **A1** that $\mathbb{U}^1, \ldots, \mathbb{U}^{\mathrm{K}}$ are drawn independently. This implies that the entries in $\mathbf{U}^1, \ldots, \mathbf{U}^{\mathrm{K}}$ keep no relation with one another. Similarly, from **A2** we know that $\mathbf{x}_{\Omega_{\mathrm{k}}}$ is drawn independently. Equivalently, $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{\mathrm{K}}$ keep no relation with one another. Intuitively, we can think of the entries in $\mathbf{U}^{\mathrm{k}}$'s and $\boldsymbol{\theta}^{\mathrm{k}}$'s as independent random numbers drawn from a distribution with full measure. We thus conclude that if $\Omega \not\subset \Omega_{\mathrm{k}}$, with probability 1 the consistency condition equation 2 would not hold.

To summarize, we know that $\mathbf{x}_{\Omega} \in \mathbb{U}^{\mathrm{k}}_{\Omega}$ if and only if there is a vector $\boldsymbol{\theta}$ that satisfies equation 1. We showed that this will be the case if and only if equation 2 holds, which in turn will happen if and only if $\Omega \subset \Omega_{\mathrm{k}}$, as claimed. □

Notice that if Ω has $\dim(\mathbb{U}^{\mathrm{k}}) + \ell$ entries, equation 1 becomes a system with $\dim(\mathbb{U}^{\mathrm{k}}) + \ell$ equations, which result in ℓ consistency conditions similar to equation 2. By the same arguments as in the proof of Lemma 2, $\mathbf{x}_{\Omega}$ will agree with $\mathbb{U}^{\mathrm{k}}_{\Omega}$ if and only if these consistency conditions hold, which will happen if and only if $\Omega \subset \Omega_{\mathrm{k}}$. We thus obtain the following corollary

Corollary 2. *Suppose **A1** and **A2** hold. Let Ω be an arbitrary subset of $[\mathrm{d}]$ with more than $\dim(\mathbb{U}^{\mathrm{k}})$ elements. Then $\mathbf{x}_\Omega \in \mathbb{U}^{\mathrm{k}}_\Omega$ if and only if $\Omega \subset \Omega_{\mathrm{k}}$.*

With this, we are ready to give a proof of Theorem 1

Proof. (Theorem 1) Let $\mathrm{r}_{\mathrm{k}} := \dim(\mathbb{U}^{\mathrm{k}})$. To identify $\Omega_1, \ldots, \Omega_{\mathrm{k}}$ we will exhaustively search for combinations Ω of $\mathrm{r}_{\mathrm{k}} + 1$ entries in $\mathbf{x}$ that match with $\mathbb{U}^{\mathrm{k}}$. By Lemma 2 we know that $\mathbf{x}_\Omega$ will match with $\mathbb{U}^{\mathrm{k}}$ if and only if $\Omega \subset \Omega_{\mathrm{k}}$. By assumption there is at least one k for which $|\Omega_{\mathrm{k}}| > \mathrm{r}_{\mathrm{k}}$, so we know that at some point we will find such a subset Ω. Once we do, we can recover $\boldsymbol{\theta}^{\mathrm{k}} = (\mathbf{U}^{\mathrm{k}}_{\Omega'})^{-1}\mathbf{x}_{\Omega'}$, where Ω' can be any subset of Ω with exactly r_{k} entries (recall that **A1** guarantees that $\mathbb{U}^{\mathrm{k}}_{\Omega'}$ is invertible). Finally, we can compute $\mathbf{x}^{\mathrm{k}} := \mathbf{U}^{\mathrm{k}}\boldsymbol{\theta}^{\mathrm{k}}$, and recover Ω_{k} by inspection (the set of zero entries in $\mathbf{x}^{\mathrm{k}} - \mathbf{x}$), where Corollary 2 guarantees that there will be no additional false matching entries. Repeating this procedure for every k we can recover $\Omega_1, \ldots, \Omega_{\mathrm{K}}$, as claimed.

To obtain the converse, suppose that r_{k} or fewer entries of $\mathbf{x}$ correspond to $\mathbb{U}^{\mathrm{k}}$ for some k. By Lemma 1, any subset Ω' of $\Omega_k \cup \Omega_0$ with $|\Omega'| \leq \mathrm{r}_{\mathrm{k}}$ satisfies $\mathbf{x}_{\Omega'} \in \mathbb{U}^{\mathrm{k}}_{\Omega'}$. Hence we can split the entries of $\mathbf{x}$ corresponding to Ω_{k} and Ω_0 in a non-unique manner, and still have them agree with $\mathbb{U}^{\mathrm{k}}$, which makes Ω_{k} unidentifiable. □

We now use the same strategy to prove Corollary 1, which extend these ideas to noisy settings.

Proof. (Corollary 1) Let $\mathbf{P}^{\mathrm{k}}_{\Omega'}$ denote the projection operator onto $\mathbb{U}^{\mathrm{k}}_{\Omega'}$. Under the assumptions of the corollary, if $\Omega' \not\subset \Omega_{\mathrm{k}}$, then with high probability (decreasing in C and increasing in c), $\|\mathbf{x}_{\Omega'} - \mathbf{P}^{\mathrm{k}}_{\Omega'}\mathbf{x}_{\Omega'}\|^2 > |\Omega'|\sigma^2$ (see Theorem 1 in (17)). In other words, the residual of projecting $\mathbf{x}_{\Omega'}$ onto $\mathbb{U}^{\mathrm{k}}_{\Omega'}$ will be too large if the entries of Ω' do not come from the kth subspace. Hence, instead of searching for a set Ω' where $\mathbf{x}_{\Omega'}$ perfectly fits in $\mathbb{U}^{\mathrm{k}}_{\Omega'}$ (as in the proof of Theorem 1), we can search for subsets $\Omega' \subset \Omega_{\mathrm{k}}$ where the residual is within the noise level, i.e., $\|\mathbf{x}_{\Omega'} - \mathbf{P}^{\mathrm{k}}_{\Omega'}\mathbf{x}_{\Omega'}\|^2 \leq |\Omega|\sigma^2$. The rest of the proof follows by the same arguments as the proof of Theorem 1. □

B Additional Experiments

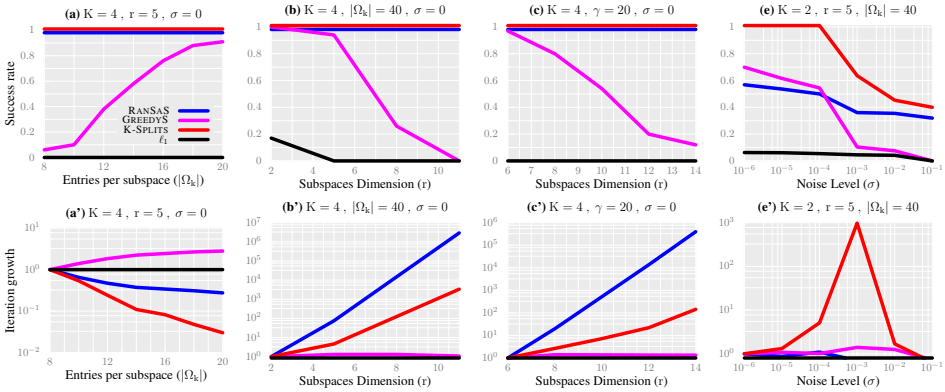

Figure 2: Complementary results to Figure 1 with $\mathrm{K} = 4$. Average success rate and iteration growth over 100 trials, as a function of the number of entries per subspace $|\Omega_{\mathrm{k}}|$ (which in turn is a proxy of the ambient dimension d), the subspaces dimensions r, the gap $\gamma := |\Omega_{\mathrm{k}}| - (\mathrm{r} + 1)$, and the noise level σ.