k-Subspaces with Missing Data

University of Wisconsin Technical Report ECE-11-02
February 15, 2011

Laura Balzano and Robert Nowak
University of Wisconsin, Madison
{sunbeam, nowak}@ece.wisc.edu

Arthur Szlam
New York University
aszlam@courant.nyu.edu

Benjamin Recht
University of Wisconsin, Madison
brecht@cs.wisc.edu

***Abstract*—Linear subspace models have recently been successfully employed to model highly incomplete high-dimensional data; they offer simplicity and ease of analysis, but they are sometimes too restrictive to model the data well. Modeling data as a union of subspaces has more flexibility while still retaining the benefits of linear modeling. This modeling approach leads to the problem of *Subspace Clustering*, or clustering vectors into groups that lie in or near the same subspace. Nearly all subspace clustering algorithms require full data, and they often rely on subspace assignment using projections: given k subspaces, a vector can be assigned to the closest subspace by calculating the projection residuals to all subspaces and choosing the one with minimum norm. This paper shows that a similar approach can be used to determine which of k subspaces is closest to a highly incomplete data vector. The approach is based on [4] and restricts the projection to observed coordinates only. We give a requirement on the angles between the vector and the various subspaces; when this requirement holds, the incomplete data vector is assigned to the correct subspace with high probability. We then discuss two implementations the k-subspaces algorithm adapted for missing data and show numerical results.**

I. Introduction

Modeling high-dimensional data with a union of subspaces is a useful generalization of subspace models [10], and has applications in machine learning, imaging, computer vision [7], and system identification [16]. An important problem which arises when modeling data with a union of subspaces is subspace clustering, or clustering vectors into groups that lie in or near the same subspace.

Many of the subspace clustering algorithms have a subroutine which assigns data vectors to subspaces based on projection residuals [15]. For example, the k-planes clustering algorithm[1] [5], [2] is an alternating minimization algorithm which alternates between estimating subspace parameters given a clustering of the data and clustering the data using projections to assign the data points to fixed subspaces.

What we call the subspace assignment subproblem can be defined as follows. Given a vector $v \in \mathbb{R}^n$ and subspaces $S^0, \ldots, S^k$, how do we determine to which subspace v is closest? Calculate the angle to each subspace using the projection:

$$\sin(\theta_i) = \frac{\|v - P_{S^i} v\|_2}{\|v\|_2} \ , \quad i = 1, \ldots, k \ ,$$

where P_{S^i} is the projection operator onto subspace S^i. A data vector is closest to the subspace corresponding to the smallest angle or projection residual norm; these are the same if numerical precision is not an issue.

In this paper we give theoretical results on this canonical subproblem with incomplete vectors. We then apply those results to an algorithm for subspace clustering with incomplete vectors. Our work is motivated by high-dimensional applications in which it is infeasible or impractical to collect complete measurements. If the data vectors lie in a single low-dimensional subspace, it is possible to apply ideas from matrix completion [3], [12], [13] for modeling and inference without complete data. These techniques have been applied to many interesting applications, including quantum state tomography [8] and most famously the Netflix challenge [1]. In this paper, we extend this work for data vectors which lie in or near a union of low-dimensional subspaces.

Thus we revise the subspace assignment subproblem to ask, given an *incomplete* data vector, how do we determine to which of k subspaces the vector is closest? We use an extension of the result in [4] for subspace detection. Our first contribution is to explore the relationship among the angles between the vector and the k subspaces. We examine how these angles impact the result of the subspace assignment, both analytically and in simulation.

The second contribution is to then develop algorithms for estimating a union of subspaces from highly incomplete data vectors. We combine k-subspaces with an algorithm developed for single subspace estimation, Grassmannian Rank-One Update Subspace Estimation [3], to address the problem of multiple subspace estimation. We describe both a batch and incremental version, and show their properties through numerical simulation.

II. Subspace Assignment

We begin by examining the problem of binary subspace assignment, i.e. $k = 2$. Let $v \in \mathbb{R}^n$ and let $S^0 \subset \mathbb{R}^n$ and $S^1 \subset \mathbb{R}^n$ be subspaces of dimension d_0 and d_1 respectively. Is v closer to S^0 or S^1? If we had complete data, we would compare the norm of the projection residual of v onto both S^0 and S^1:

$$\|v - P_{S^0} v\|_2^2 \overset{?}{<} \|v - P_{S^1} v\|_2^2 \ . \tag{1}$$

[1] This is also called k-subspaces in the literature, and we use the terminology interchangeably.

If this inequality holds we would assign v to S^0; otherwise we would assign v to S^1.

Now consider the situation when we only observe a set $\Omega \subset \{1, \ldots, n\}$ of indices of v. Denote the observed vector as v_Ω. Let the columns of an orthonormal matrix U span the d-dimensional subspace $S \in \mathbb{R}^n$. Then the projection operator restricted to Ω is defined as

$$P_{S_\Omega} = U_\Omega \left(U_\Omega^T U_\Omega\right)^{-1} U_\Omega^T \ . \tag{2}$$

The notation U_Ω denotes a restriction to the rows of U indicated by the set Ω. We base our subspace assignment on this projection residual:

$$\|v_\Omega - P_{S^0_\Omega} v_\Omega\|_2^2 \stackrel{?}{<} \|v_\Omega - P_{S^1_\Omega} v_\Omega\|_2^2 \ . \tag{3}$$

In what follows we show that with enough observations, the subspace assignment based on (3) will be the same as that for (1) with high probability.

A. Results

We first restate a theorem from [4]. It provides us with a relationship between the residual of a complete-data projection of v and the projection of v restricted to the observed entries. Let $v = x + y$, where $x \in S$ and $y \in S^\perp$.

Theorem 1. *Let $\delta > 0$ and $m \geq \frac{8}{3}\mu(S) d \log\left(\frac{2d}{\delta}\right)$. Then with probability at least $1 - 3\delta$,*

$$\frac{m(1-\alpha) - d\mu(S)\frac{(1+\beta)^2}{(1-\gamma)}}{n}\|v - P_S v\|_2^2 \leq \|v_\Omega - P_{S_\Omega} v_\Omega\|_2^2$$

and with probability at least $1 - \delta$,

$$\|v_\Omega - P_{S_\Omega} v_\Omega\|_2^2 \leq (1+\alpha)\frac{m}{n}\|v - P_S v\|_2^2$$

where $\alpha = \sqrt{\frac{2\mu(y)^2}{m}\log\left(\frac{1}{\delta}\right)}$, $\beta = \sqrt{2\mu(y)\log\left(\frac{1}{\delta}\right)}$, and $\gamma = \sqrt{\frac{8d\mu(S)}{3m}\log\left(\frac{2d}{\delta}\right)}$.

Now define the angle between the vector v and its projection into the two subspaces S^0, S^1 as θ_0 and θ_1:

$$\theta_0 = \sin^{-1}\left(\frac{\|v - P_{S^0} v\|_2}{\|v\|_2}\right) \tag{4}$$

$$\theta_1 = \sin^{-1}\left(\frac{\|v - P_{S^1} v\|_2}{\|v\|_2}\right) \tag{5}$$

Let $v = x_0 + y_0 = x_1 + y_1$, where $x_0 \in S^0$, $y_0 \perp S^0$, $x_1 \in S^1$, and $y_1 \perp S^1$. For notational simplicity and without loss of generality we focus on the situation when $\theta_0 < \theta_1$ and define

$$C(m) = \frac{m(1-\alpha_1) - d_1\mu(S^1)\frac{(1+\beta_1)^2}{(1-\gamma_1)}}{m(1+\alpha_0)} \ , \tag{6}$$

where α_1, β_1, and γ_1 are defined as in Theorem 1 using d_1, $\mu(y_1)$ and $\mu(S^1)$, and α_0 is defined using $\mu(y_0)$. Notice that $C(m) \nearrow 1$ as $m \to \infty$.

Theorem 2. *Let $\delta > 0$ and $m \geq \frac{8}{3} d_1 \mu(S^1) \log\left(\frac{2d_1}{\delta}\right)$. Assume that*

$$\sin^2(\theta_0) < C(m)\sin^2(\theta_1) \ . \tag{7}$$

Then with probability at least $1 - 4\delta$,

$$\|v_\Omega - P_{S^0_\Omega} v_\Omega\|_2^2 < \|v_\Omega - P_{S^1_\Omega} v_\Omega\|_2^2 \ .$$

Before the proof we consider consequences of the theorem. First we consider the situation where $\theta_0 = 0$, i.e., the vector v is *in* the hypothesized subspace S^0. As long as $\theta_1 \neq 0$, the ratio $\sin^2(\theta_0)/\sin^2(\theta_1) = 0$. This in turn implies that the number of observations required does not depend on θ_1 nor on the relationship of θ_0 to θ_1, and the condition (7) is simply that $C(m) > 0$. To guarantee $\theta_1 \neq 0$ for arbitrary $v \in S^0$, we must have that S^0 and S^1 are linearly independent[2]. In other words, if S^0 and S^1 are linearly independent, and the vector is in either S^0 or S^1, the number of observations to guarantee the test works does not depend on the angle of v to the other subspace.

If, on the other hand, S^0 and S^1 are not linearly independent, there are vectors in the two subspaces which are arbitrarily close to one another; for any fixed m there exists a vector in S^0 for which the incomplete data projection residual would not be valid.

Now we consider the situation where v is not *in* the subspace, but is simply *closer*: $0 < \theta_0 < \theta_1$. Thus $\sin^2(\theta_0)/\sin^2(\theta_1) > 0$. As the gap $\theta_1 - \theta_0$ decreases, $\sin^2(\theta_0)/\sin^2(\theta_1) \nearrow 1$. Consequently, as this gap narrows, we must increase m to guarantee that the subspace assignment based on (3) gives the same result as that of (1).

Proof: From Theorem 1 and the union bound, the following two statements hold simultaneously with probability at least $1 - 4\delta$:

$$\|v_\Omega - P_{S^0_\Omega} v_\Omega\|_2^2 \leq (1+\alpha_0)\frac{m}{n}\|v - P_{S^0} v\|_2^2$$

and

$$\frac{m(1-\alpha_1) - d_1\mu(S^1)\frac{(1+\beta_1)^2}{(1-\gamma_1)}}{n}\|v - P_{S^1} v\|_2^2 \leq \|v_\Omega - P_{S^1_\Omega} v_\Omega\|_2^2 \ .$$

Thus if

$$\|v - P_{S^0} v\|_2^2 < C(m)\|v - P_{S^1} v\|_2^2 \ , \tag{8}$$

we have the conclusion of the theorem. But using (4) and (5), this statement is equivalent to our requirement that

$$\sin^2(\theta_0) < C(m)\sin^2(\theta_1) \ ,$$

completing the proof. ∎

This result can be directly extended to the situation where there are multiple subspaces S^i, $i = 0, \ldots, k-1$. Again without loss of generality we focus on the situation where $\theta_0 < \theta_i$, $\forall i$, and define

$$C_i(m) = \frac{m(1-\alpha_i) - d_i\mu(S^i)\frac{(1+\beta_i)^2}{(1-\gamma_i)}}{m(1+\alpha_0)} \ ,$$

[2]Two subspaces are linearly independent if the dimension of their union is equal to the sum of their dimensions.

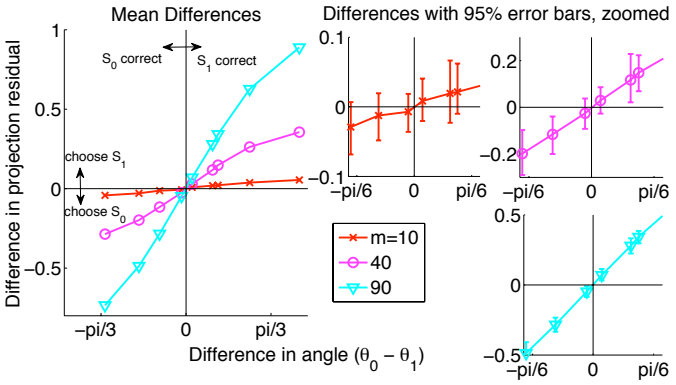

Fig. 1. Simulation results for a binary subspace assignment with $n = 100$, $d_0 = 5$, $d_1 = 5$. Projection Residual Difference is defined as $\|v_\Omega - P_{S^0_\Omega} v_\Omega\|_2^2 - \|v_\Omega - P_{S^1_\Omega} v_\Omega\|_2^2$; thus, S^0 is the chosen subspace when the residual is negative. The curves shown are averaged over 100 random sample sets of various sizes as denoted by the legend. The right three plots are zoomed around $\theta_0 - \theta_1 = 0$; 95% confidence intervals are shown.

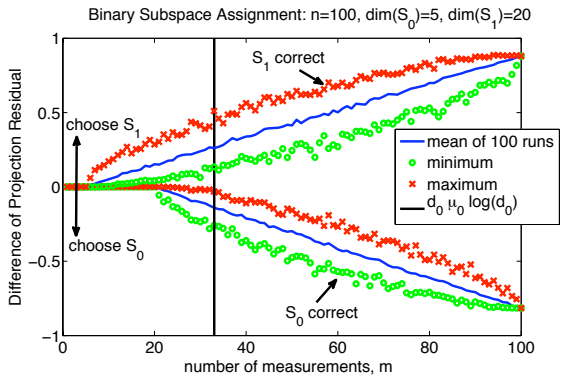

Fig. 2. Simulation results for a binary subspace assignment with $n = 100$, $d_0 = 5$, $d_1 = 20$. The upper curves represent a case where the angle difference $\theta_0 - \theta_1 = 1.078$ degrees, thus resulting in S_1 being the correct choice. The lower curves represent a case where the angle difference $\theta_1 - \theta_0 = 1.083$ degrees, thus resulting in S^0 being the correct choice.

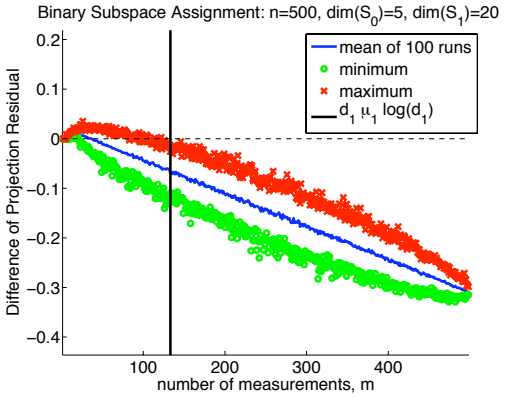

Fig. 3. Simulation results for a binary subspace assignment with $n = 500$, $d_0 = 5$, $d_1 = 20$. The angle difference $\theta_1 - \theta_0 = 0.3156$ degrees, thus S^0 is the correct choice. The approximate measurement requirement $d_1 \mu_1 \log(d_1)$ is the black line shown; beyond it the worst-case residual difference is in favor of S^0.

where α_i, β_i, and γ_i are defined as in Theorem 1 using d_i, $\mu(y_i)$ and $\mu(S^i)$, and α_0 is defined using $\mu(y_0)$.

Corollary 1. *Let* $m \geq \frac{8}{3} max_{i \neq 0} \left(d_i \mu(S^i) \log\left(\frac{2d_i}{\delta}\right)\right)$ *for fixed* $\delta > 0$. *Assume that*

$$\sin^2(\theta_0) < C_i(m) \sin^2(\theta_i) \ , \quad \forall i \neq 0 \ .$$

Then with probability at least $1 - 4(k-1)\delta$,

$$\|v_\Omega - P_{S^0_\Omega} v_\Omega\|_2^2 < \|v_\Omega - P_{S^i_\Omega} v_\Omega\|_2^2 \ , \quad \forall i \neq 0 \ .$$

B. Numerical Analysis

In order to illustrate the output of the test given by (3), we show its behavior in simulation. First we consider the most basic scenario where both subspaces are of the same dimension. Figure 1 shows the behavior of the comparison of projection residuals depending on the angle difference, $\theta_0 - \theta_1$. As the angle difference nears zero, even nearly complete vectors do not always result in the correct subspace being chosen.

Next we consider a scenario where the subspaces are of different dimension. In Figures 2 and 3 we examine the difference of the projection residual as m increases, for unbalanced subspace dimensions of $d_0 = 5$ and $d_1 = 20$. Both plots have a fixed vector v at a fixed angle to both subspaces, but the number of observations m is varied. Figure 2 shows the projection residual difference for two vectors v; one which is closer to S^0 and one which is closer to S^1. As is evident, the vector nearer S_1 can be distinguished with a small number of observations. However, the vector closer to S^0 looks as though it may be near either subspace until we have more than 20 observations.

Finally, Figure 3 shows a similar scenario but with a smaller difference in angle and a larger ambient dimension. Here were highlight the behavior of the residual near the measurement cutoff $d_1 \mu_1 log(d_1)$. Though the mean residual difference, averaged over many possible sample sets, is almost always below zero, the worst-case difference $\|v_\Omega - P_{S^0_\Omega} v_\Omega\|_2^2 - \|v_\Omega - P_{S^1_\Omega} v_\Omega\|_2^2$ is often positive, in which case we would incorrectly select S^1.

III. Subspace Clustering

The GROUSE algorithm [3], or Grassmannian Rank-One Update Subspace Estimation, was developed to do single subspace estimation with highly incomplete data vectors. Armed with Theorem 2, we can combine k-subspaces and GROUSE to multiple subspace estimation.

The standard k-subspaces algorithm is described in [5], [2]; we simply replace certain steps of the standard algorithm and its incremental version with GROUSE to allow for flexibility when observations are incomplete. Our algorithms currently require knowledge of the number of subspaces k and their dimensions. Our simulations show scenarios where the subspaces are of the same dimension, but the algorithms do not require this.

Algorithm 1 k-subspaces with the GROUSE: incremental

Require: A collection of vectors $v_\Omega(t)$, $t = 1, \ldots, T$, and the observed indices $\Omega(t)$. An integer number of subspaces k and dimensions d_i, $i = 1, \ldots, k$. A maximum number of iterations, maxIter. A fixed step size η.

1: **Initialize Subspaces:** Zero-fill the vectors and collect them in a matrix V. Initialize k subspace estimates using probabilistic farthest insertion.
2: **Calculate Orthonormal Bases** U_j, $j = 1, \ldots, k$. Let $Q_{j_\Omega} = (U_{j_\Omega}^T U_{j_\Omega})^{-1} U_{j_\Omega}^T$
3: **for** $i = 1, \ldots,$ maxIter **do**
4: **Select a vector at random,** v_Ω.
5: **for** $j = 1, \ldots, k$ **do**
6: **Calculate weights of projection onto** j^{th} **subspace:** $w(j) = Q_{j_\Omega} v_\Omega$.
7: **Calculate Projection Residuals to** j^{th} **subspace:** $r(j) = \|v_\Omega - U_{j_\Omega} w(j)\|_2^2$.
8: **end for**
9: **Select min residual:** $\widehat{j} = \text{argmin}_j\, r(j)$. Set $r = r(\widehat{j})$ and $w = w(\widehat{j})$. Define $p = v_0 - r$, where v_0 is zero-filled v_Ω.
10: **Update subspace**:
$$U_{\widehat{j}} = U_{\widehat{j}} + \left((\cos(\sigma\eta) - 1) \frac{p}{\|p\|} + \sin(\sigma\eta) \frac{r}{\|r\|} \right) \frac{w^T}{\|w\|}$$
where $\sigma = \|r\|\|p\|$
11: **end for**

To initialize the subspaces we use a version of probabilistic farthest insertion, as in [11], modified for missing data by simply zero-filling the unobserved entries in each vector and collecting them in a matrix V. Specifically, we pick a random point $v_0 \in V$. We then calculate the $d + q$ nearest neighbors to v_0, where q is a nonnegative parameter[3], and calculate the best fit subspace S^0 to the neighborhood of v_0. Next we choose another random point with probability proportional to the distance $dist(v, S^0)^2$, and find the best fit subspace S^1 of its $d + q$ neighborhood. For j^{th} neighborhood, we pick the center with probability proportional to $\min(dist(v, S^0)^2, ..., dist(v, S^{j-1})^2)$.

To refine the initial subspaces, we use GROUSE [3] adapted in two ways to multiple subspaces: one batch and one incremental. The batch version simply uses GROUSE subspace estimation in place of the SVD step for subspace estimation used in the standard k-subspaces algorithm [5], [2], [15]. Given a cluster of vectors to incorporate into S^j, and given the step size η, several steps of incremental gradient descent are performed with a diminishing step size to update the subspace estimate [3]. We note that any matrix completion algorithm that finds the column space, such as those found in [14], [9], [6], could also be used in this step with missing data.

The incremental version in Algorithm 1 is a form of sequential k-means adapted to k subspaces. We note that this algorithm works as written for the case when the data vector is complete; this is a novel approach to the full-data k subspaces clustering problem, and offers benefits in terms of computational complexity. We draw a sample v_Ω from the data vectors uniformly at random and find its closest subspace $\widehat{j} = \text{argmin}_j\, dist(v, S^j)$. $S^{\widehat{j}}$ is updated by a gradient step with respect to v_Ω of a fixed size η; we use a geodesic step on the set of orthogonal matrices as in [3], as given explicitly by Step 10 in Algorithm 1. This process is repeated maxIter steps, or until some convergence criterion is met. The incremental algorithm requires less computation than the batch algorithm, and GROUSE is specially suited for incremental updates.

At this point we note that if the sum of the dimensions of the subspaces $D := \sum_{j=1}^k d_j$ is significantly less than the ambient dimension n, a very simple two-stage approach to subspace clustering is to first perform matrix completion on the data matrix V to recover a rank D matrix whose columns will span the union of subspaces. With these complete data, any standard subspace clustering algorithm can be applied. However, there are two situations when this is not possible. First, it may be that D is actually greater than or equal to n, if there are many subspaces of small dimension. We explore this scenario in Figure 4. Second, we may have collected the $d \log(d)$ observations per vector which are sufficient for subspace assignment, but not $D \log(n)$ observations which are sufficient for matrix completion. Simulation results for this scenario are shown in Figure 5.

A. *Subspace Clustering Simulations*

We show the results of three simulation scenarios. The first is the simplest: the data vectors come from subspaces which are orthogonal, and the sum of the dimensions of the subspaces is the ambient dimension: $D = n$. The left plot of Figure 4 shows the results. The ambient dimension $n = 20$, the intrinsic dimension of each subspace $d = 5$, and there are 4 subspaces. The data matrix V consists of 50 vectors per subspace. The parameter for nearest neighbor subspace estimation is $q = 5$.

The error is calculated as compared to ground truth: we examine the cluster assignments of a set vectors who share the same true cluster. Whichever cluster ID has the largest number of vectors assigned to it, we consider those to be clustered correctly; we count the rest as incorrectly clustered vectors. We note that this error can be minimized trivially by an algorithm which assigns all the vectors to one cluster; however our algorithms also minimize Euclidean distance to low dimensional subspaces, and we have verified that the clusters are evenly sized. Mathematically, let A_j, $j = 1, \ldots, k$ be sets of indices corresponding to ground-truth cluster assignments. Let B_j be the cluster assignments chosen by the algorithm. For $l = 1, \ldots, k$, we find

$$\widehat{j}_l = \text{argmax}_j |B_j \cap A_l| \,,$$

where $|\cdot|$ denotes the cardinality of a set. Then the error is

$$\sum_{l=1}^{k} |A_l \backslash \{A_l \cap B_{\widehat{j}_l}\}| \quad .$$

Thus in the left plot of Figure 4, the worst possible result is 150 incorrectly clustered vectors. Both algorithms perform

[3]For methods of automatically picking q, see [17].

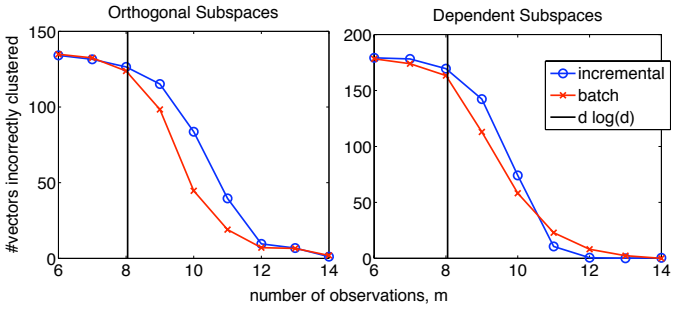

Fig. 4. Simulation results for subspace clustering. On the left we have $n = 20$, $d = 5$, $k = 4$ and orthogonal subspaces. On the right we have $k = 5$ and thus linearly dependent subspaces. The error measure is defined in Section III-A. The curves shown are averaged over 100 random observation sets, with observation set size varied along the x-axis.

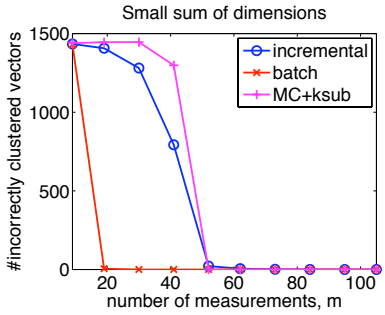

Fig. 5. For this simulation, $n = 150$, $d = 5$, $k = 6$, and the dimension of the union of subspaces is $D = 30$. We compare our missing data subspace clustering algorithms to a two-stage approach: matrix completion is first used to complete the data matrix of rank 30, and then k-planes clustering is used for clustering. Results are averaged over 10 random observation sets.

well with a little more than $2d$ entries observed. The batch algorithm performs slightly better in mid-ranges of samples. However the computation time, which we do not discuss for lack of space, is much smaller for the incremental algorithm.

The second simulation scenario is a case where the sum of the dimensions of the subspaces is greater than the ambient dimension, $D > n$. These results can be seen in the right hand plot of Figure 4. The simulation parameters are $n = 20$, $d = 5$, $k = 5$, $q = 5$, and the data matrix consists of 50 vectors per subspace; so now the worst possible error is 200 incorrectly clustered vectors. Again both algorithms perform well for a little more than $2d$ of the entries observed.

The third scenario is one where sum of the dimensions of the subspaces is less than the ambient dimension, $D < n$. Here we compare to the two-stage approach of matrix completion plus full-data k-planes clustering. Once there are enough measurements to estimate the rank-D matrix, all the algorithms perform with zero error. However in the low-observation regime, there are still enough measurements to estimate each of the k subspaces, and both the batch and incremental algorithms outperform matrix completion followed by k-planes clustering.

IV. Conclusion

Clustering vectors into subspaces is a problem with many applications in machine learning and signal processing where it may be impossible to collect a complete matrix of observations. Subspace clustering algorithms rely on the fact that, from a set of subspaces S^j, $j = 1, \ldots, k$, the closest subspace to a vector v can be found by calculating projections to all the subspaces and choosing the one with minimum projection residual. In this paper we showed that this is also possible with missing data if one compares the incomplete data projection residual as in (3). We derived the number of observations sufficient to guarantee that this closest subspace assignment is the same as that given by the full-data comparison. The number of measurements depends on the angles from v to each of the subspaces. We applied this result and developed a subspace clustering algorithm with missing data; we showed that it performs well in various scenarios, even when the sum of dimensions of the subspaces is greater than or equal to the ambient dimension.

References

[1] ACM SIGKDD and Netflix. *Proceedings of KDD Cup and Workshop*, 2007. Proceedings available online at http://www.cs.uic.edu/~liub/KDD-cup-2007/proceedings.html.

[2] P. K. Agarwal and N. H. Mustafa. k-Means projective clustering. In *Proceedings of ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems*, 2004.

[3] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proceedings of the Allerton conference on Communication, Control, and Computing*, 2010.

[4] L. Balzano, B. Recht, and R. Nowak. High-dimensional matched subspace detection when data are missing. In *Proceedings of ISIT*, June 2010.

[5] P. S. Bradley and O. L. Mangasarian. k-Plane clustering. *Journal of Global Optimization*, 16:23–32, 2000.

[6] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2008.

[7] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29, 1998.

[8] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. Preprint available at http//arxiv.org/abs/0909.3304, 2009.

[9] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, June 2010.

[10] Y. M. Lu and M. N. Do. A theory for sampling signals from a union of subspaces. *IEEE Transactions on Signal Processing*, 56(6), June 2008.

[11] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *In 47th IEEE Symposium on the Foundations of Computer Science (FOCS*, pages 165–176, 2006.

[12] B. Recht. A simpler approach to matrix completion. 2009. Preprint available at http://arxiv.org/abs/0910.0651.

[13] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization. *SIAM Review*, 2007. To appear. Preprint Available at http://pages.cs.wisc.edu/~brecht/publications.html.

[14] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Preprint available at http://www.optimization-online.org/DB_HTML/2009/03/2268.html, 2009.

[15] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 2010.

[16] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proceedings of the Conference on Decision and Control*, pages 167–172, 2003.

[17] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Randomized hybrid linear modeling by local best fit flats. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 2010.