K-SUBSPACES WITH MISSING DATA

Laura Balzano[1]*, Arthur Szlam*[2]*, Benjamin Recht*[1]*, Robert Nowak*[1]

[1]University of Wisconsin, Madison; [2]The City University of New York

ABSTRACT

Linear subspace models have recently been successfully employed to model highly incomplete high-dimensional data, but they are sometimes too restrictive to model the data well. Modeling data as a union of subspaces gives more flexibility and leads to the problem of *Subspace Clustering*, or clustering vectors into groups that lie in or near the same subspace. Low-rank matrix completion allows one to estimate a single subspace from incomplete data, and this work has recently been extended for the union of subspaces problem [3]. However, the algorithm analyzed there is computationally demanding. Here we present a fast algorithm that combines GROUSE, an incremental matrix completion algorithm, and k-subspaces, the alternating minimization heuristic for solving the subspace clustering problem. k-GROUSE is two orders of magnitude faster than the algorithm proposed in [3] and relies on a slightly more general projection theorem which we present here.

***Index Terms*—** Union of subspaces; subspace clustering; matrix completion

1. INTRODUCTION

Modeling high-dimensional data with a union of subspaces is a useful generalization of a single low-rank subspace model. Taking a collection of data points which lie in a union of subspaces, and identifying which points share the same subspace, is called *Subspace Clustering* and arises in machine learning, imaging, computer vision [8], and system identification [15]. Correct algorithms for subspace clustering are now known, either with noise, outliers, or missing data [3, 7, 9, 14, 16]. These algorithms may be tractable but are not always as fast as is needed for massive data applications.

We focus on the missing data problem. In countless high-dimensional applications, it is impossible or too costly to obtain every complete data point. For example, for internet tomography it is problematic to try to actively probe every IP address [3]; also particular SIFT features of a video sequence may be obstructed in certain frames. Thus one may wish to apply recent results from matrix completion [4, 11] for modeling and inference without complete data. These techniques have been applied to many interesting applications, including most famously the Netflix challenge [1].

Recent work [3] has provably extended matrix completion results to solve the problem of subspace clustering with incomplete data. However, the algorithm analyzed in that paper is computationally burdensome. In the present paper we introduce k-GROUSE, an algorithm which empirically achieves the correct subspace clustering and estimation in two orders of magnitude less time.

k-GROUSE is a combination of GROUSE [4] and k-subspaces. GROUSE, or Grassmannian Rank-One Subspace Estimation, is a subspace estimation algorithm for incomplete data vectors. k-subspaces is an alternating heuristic: starting with some initial subspaces, vectors are clustered by *subspace assignment* (which we define below); subspaces are then re-estimated and the process is repeated until convergence. k-GROUSE updates each subspace incrementally, one vector at a time. We compare k-GROUSE to a batch version and the algorithm of [3] through numerical simulation.

What we call the subspace assignment subproblem can be defined as follows. Given a vector $v \in \mathbb{R}^n$ and subspaces $S^0, \ldots, S^k$, we wish to determine the subspace closest to v. Calculate the residual to each subspace using the projection: $\|v - P_{S^i} v\|_2^2$, $i = 1, \ldots, k$, where P_{S^i} is the projection operator onto subspace S^i. A data vector is closest to the subspace corresponding to the smallest projection residual norm.

In this paper we give theoretical results on this canonical subproblem with incomplete vectors: given an *incomplete* data vector, how do we determine to which of k subspaces the vector is closest? This results in a generalization of a theorem in [3] which guarantees correct assignment for a vector in one of the subspaces (e.g. $v \in S^1$).

2. SUBSPACE ASSIGNMENT

We begin by examining the problem of binary subspace assignment, i.e. $k = 2$. Let $v \in \mathbb{R}^n$ and let $S^0, S^1 \subset \mathbb{R}^n$ 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:

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

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

The Wisconsin authors would like to acknowledge the generous support of AFOSR FA9550-09-1-0140, and AD Szlam from NSF DMS-1201666.

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 we define the projection operator restricted to Ω as $P_{S_\Omega} = U_\Omega \left(U_\Omega^T U_\Omega\right)^{-1} U_\Omega^T$, where 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:

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

In what follows we show that with enough observations, the subspace assignment based on (2) will be the same as that for (1) with high probability.

2.1. Results

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

$$\theta_0 = \sin^{-1}\left(\frac{\|v - P_{S^0} v\|_2}{\|v\|_2}\right) \tag{3}$$

and θ_1 is defined similarly. Following the notation of [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 $y_1 \perp S^1$. Let $\mu(S) = \frac{n}{d} max_j \|P_S e_j\|_2^2$, where e_j is the j^{th} canonical basis vector. Let $m := |\Omega|$ and choose a $\delta > 0$ as a confidence parameter. For notational simplicity and without loss of generality we focus on the situation when $\theta_0 < \theta_1$ and define

$$C(m) = \frac{m(1-\alpha_1) - d_1 \mu(S^1) \frac{(1+\beta_1)^2}{(1-\gamma_1)}}{m(1+\alpha_0)} \,, \tag{4}$$

where $\alpha_1 = \sqrt{\frac{2\mu(y_1)^2}{m} \log\left(\frac{1}{\delta}\right)}$, $\beta_1 = \sqrt{2\mu(y_1) \log\left(\frac{1}{\delta}\right)}$, $\gamma_1 = \sqrt{\frac{8 d_1 \mu(S^1)}{3m} \log\left(\frac{2d_1}{\delta}\right)}$, and $\alpha_0 = \sqrt{\frac{2\mu(y_0)^2}{m} \log\left(\frac{1}{\delta}\right)}$.

Notice that $C(m) \nearrow 1$ as $m \to \infty$.

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

$$\sin^2(\theta_0) < C(m) \sin^2(\theta_1) \,. \tag{5}$$

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. This particular case was proved in [3]. 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 (5) 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[1]. 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 (2) gives the same result as that of (1).

Proof. From Theorem 1 of [5] 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

$$\|v - P_{S^0} v\|_2^2 < C(m) \|v - P_{S^1} v\|_2^2 \,, \tag{6}$$

we have the conclusion of the theorem. But using (3), 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. Again without loss of generality we focus on the situation where $\theta_0 < \theta_i$, $\forall i$, and define $C_i(m) = \frac{m(1-\alpha_i) - d_i \mu(S^i)\frac{(1+\beta_i)^2}{(1-\gamma_i)}}{m(1+\alpha_0)}$, where α_i, β_i, and γ_i are defined as in Theorem 1 using d_i, $\mu(y_i)$ and $\mu(S^i)$.

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 $1 - 4(k-1)\delta$,

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

3. SUBSPACE CLUSTERING

The GROUSE algorithm [4], or Grassmannian Rank-One Update Subspace Estimation, was developed to do single subspace estimation with highly incomplete data vectors. Armed with Theorem 1, we can combine k-subspaces and GROUSE to multiple subspace estimation. The standard k-subspaces algorithm is described in [6, 2]; our version is incremental and based on GROUSE to allow for flexibility when observations are incomplete. Here we will consider the case of subspaces of equal dimension.

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

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 dimension d. A maximum number of iterations, maxIter. A fixed step size η.

Initialize Subspaces: Zero-fill the vectors and collect them in a matrix V. Initialize k subspace estimates using probabilistic farthest insertion.

Calculate Orthonormal Bases U_j, $j = 1, \ldots, k$. Let $Q_{j_\Omega} = (U_{j_\Omega}^T U_{j_\Omega})^{-1} U_{j_\Omega}^T$

for $i = 1, \ldots,$ maxIter **do**

  Select a vector at random, v_Ω.

  for $j = 1, \ldots, k$ **do**

    Calculate projection weights: $w(j) = Q_{j_\Omega} v_\Omega$.

    Calculate residual: $r(j) = \|v_\Omega - U_{j_\Omega} w(j)\|_2^2$.

  end for

  Select min residual: $\widehat{j} = \text{argmin}_j\, r(j)$. Set $r = r(\widehat{j})$, $w = w(\widehat{j})$, $p = v_0 - r$, where v_0 is zero-filled v_Ω.

  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 $\sigma = \|r\|\|p\|$

end for

To initialize the subspaces we use a version of probabilistic farthest insertion, as in [10], 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 as the first cluster "center," $v_0 \in V$. We then calculate the $d + q$ nearest neighbors to v_0, where d is the subspace dimension and q is a nonnegative parameter [17], and calculate the best fit subspace S^0 to the neighborhood of v_0. For the next center 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, the incremental algorithm k-GROUSE is presented in Algorithm 1. It is a form of sequential k-means adapted to k subspaces. In each iteration, a single incomplete vector v_Ω is chosen, the closest subspace is found by using Corollary 1, and this subspace is updated via GROUSE with the data vector v_Ω. This is repeated several times until some criteria are met. We note that this algorithm works as written for the case when the data vector is complete.

With matrix completion in mind, one may also consider a batch version of k-subspaces. The batch version would simply use any matrix completion algorithm, such as GROUSE or the one found in [12], in place of the SVD step for subspace estimation used in the standard k-subspaces algorithm [6, 2, 13]. Given a cluster of vectors, matrix completion would be performed to get a subspace estimate; then vectors would be reassigned, and the process repeated.

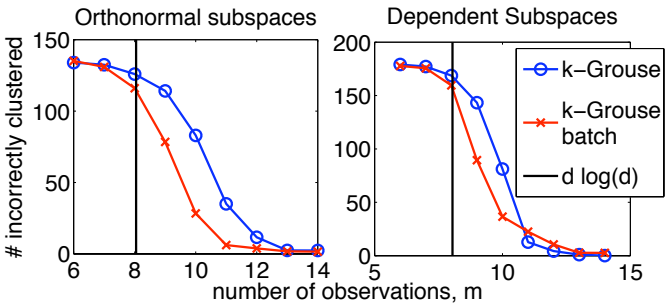

Fig. 1. Simulation results: On the left we have $n = 20$, $d = 5$, $k = 4$ ($D = n$) and orthogonal subspaces. On the right we have $k = 5$ ($D > n$) and thus linearly dependent subspaces. The error measure is defined in Section 3.1. The curves shown are averaged over 100 random observation sets.

As written, k-GROUSE and this suggested batch version both require knowledge of the number of subspaces k and their dimensions, whereas the algorithm in [3] only requires an upper bound on both values. Our simulations show scenarios where the subspaces are of the same dimension, but the algorithms do not require this.

At this point we note that if the sum of the dimensions of the subspaces $D := kd$ is significantly less than n, a two-stage approach to subspace clustering is to first perform matrix completion to recover a rank D matrix and then apply any full-data subspace clustering algorithm; this is not possible if D is large compared to n; See Figure 1. Even if D is relatively small, 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. This is the setup of the simulation of Table 1.

3.1. Subspace Clustering Simulations

We show the results of three simulation scenarios. In the first, 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 1 shows the results. The data matrix consists of $N = 200$ points for the left plot and $N = 300$ points for the right point, i.e. 50 vectors per subspace. The parameter for nearest neighbor subspace estimation is $q = 5$.

The error is calculated as compared to ground truth. 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 $\sum_{l=1}^k |A_l \backslash \{A_l \cap B_{\widehat{j}_l}\}|$. We note that this error can be minimized trivially by an algorithm which assigns all the vectors to one cluster; however these algorithms also minimize distance to low dimensional subspaces, and we have verified that the clusters are about the correct size. Results for

$D > n$ can be seen in the right hand plot of Figure 1.

	Computation Time (sec)		% successful
Algorithm	average	std. dev.	trials
Alg from [3]	10395.0	655.8	56
Batch k-Sub	1079.5	17.8	97
Alg 1	127.6	0.24	93

Table 1. The problem size is $n = 50$, $k = 10$, $d = 4$, ($D < n$) and $N = 40,000$. 60% of the entries were sampled. The algorithm in [3] took 2.5 hours whereas Algorithm 1 took 2 minutes. Successful trials are those in which the clustering had no errors. This percentage is low for [3] due to an insufficient number of seed points; increasing the number to achieve high accuracy triples the running time.

The main benefit of k-GROUSE is its speed. The algorithm in [3] requires that the number of matrix columns $N = n^p$ for some $p \geq 2$ in order to guarantee that local neighborhoods can be found despite missing data. Then distances must be computed between $k \log k$ seed columns and all these N columns using a mask, unique for every pair, that identifies the shared observations between those two columns. k-GROUSE on the other hand uses the rough initialization using zero-filled distances, and then each incremental update only takes $O(kmd^2 + nd)$ time. In Table 1 we show results of simulations run in Matlab on a Dell Precision T5500n with a Dual Quad Core Intel Xeon 2.53GHz processor and 12 GB of RAM. Clearly k-GROUSE far outpaces the algorithm in [3]. More importantly, it even performs an order of magnitude faster than the batch heuristic algorithm.

4. 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. In this paper we showed that it is possible to assign incomplete vectors to subspaces, and we presented a fast algorithm for subspace clustering with missing data.

5. 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 Symp. on Princ. of database systems*, 2004.

[3] L. Balzano, B. Eriksson, and R. Nowak. High rank matrix completion and subspace clustering with missing data. In *Proceedings of the conference on Artificial Intelligence and Statistics (AIStats)*, 2012.

[4] 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.

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

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

[7] G. Chen and M. Maggioni. Multiscale Geometric and Spectral Analysis of Plane Arrangements. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, June 2011.

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

[9] G. Lerman and T. Zhang. Robust Recovery of Multiple Subspaces by L_p Minimization. *Annals of Statistics*, 39(5):2686–2715, 2011.

[10] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *In FOCS*, pages 165–176, 2006.

[11] B. Recht. A simpler approach to matrix completion. *Jrnl. of Machine Learning Rsrch.*, 12:3413–3430, 2011.

[12] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *SIAM J. Imaging Sci.*, 4:573 – 596, 2011.

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

[14] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005.

[15] 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.

[16] Y.-L. Yu and D. Schuurmans. Rank/norm regularization with closed-form solutions: Application to subspace clustering. In *Conference on Uncertainty in Artificial Intelligence*, 2011.

[17] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Randomized hybrid linear modeling by local best fit flats. In *CVPR*, San Francisco, CA, June 2010.