On GROUSE and Incremental SVD

Laura Balzano
University of Michigan
girasole@umich.edu

Stephen J. Wright
University of Wisconsin, Madison
swright@cs.wisc.edu

***Abstract*—GROUSE (Grassmannian Rank-One Update Subspace Estimation) [1] is an incremental algorithm for identifying a subspace of $\mathbb{R}^n$ from a sequence of vectors in this subspace, where only a subset of components of each vector is revealed at each iteration. Recent analysis [2] has shown that GROUSE converges locally at an expected linear rate, under certain assumptions. GROUSE has a similar flavor to the incremental singular value decomposition algorithm [4], which updates the SVD of a matrix following addition of a single column. In this paper, we modify the incremental SVD approach to handle missing data, and demonstrate that this modified approach is equivalent to GROUSE, for a certain choice of an algorithmic parameter.**

I. INTRODUCTION

Subspace estimation and singular value decomposition have been important tools in linear algebra and data analysis for several decades. They are used to understand the principal components of a signal, to reject noise, and to identify best approximations.

The GROUSE (Grassmannian Rank-One Update Subspace Estimation) algorithm, described in [1], aims to identify a subspace of low dimension, given data consisting of a sequence of vectors in the subspace that are missing many of their components. Missing data is common in such big-data applications as low-cost sensor networks (in which data often get lost from corruption or bad communication links), recommender systems (where we are missing consumers' opinions on products they have yet to try), and health care (where a patient's health status is only sparsely sampled in time). GROUSE was developed originally in an online setting, to be used with streaming data or when the principal components of the signal may be time-varying. Several subspace estimation algorithms in the past [6] have also been developed for the online case and have even used stochastic gradient, though GROUSE and the approach described in [3] are the first to deal with missing data.

Recent developments in the closely related field of matrix completion have shown that low-rank matrices can be reconstructed from limited information, using tractable optimization formulations [5], [7]. Given this experience, it is not surprising that subspace identification is possible even when the revealed data is incomplete, under appropriate incoherence assumptions and using appropriate algorithms.

GROUSE maintains an $n \times d$ matrix with orthonormal columns that is updated by a rank-one matrix at each iteration. The update strategy is redolent of other optimization approaches such as gradient projection, stochastic gradient, and quasi-Newton methods. It is related also to the incremental singular value decomposition approach of [4], in which the SVD of a matrix is updated inexpensively after addition of a column. We aim in this note to explore the relationship between the GROUSE and incremental SVD approaches. We show that when the incremental SVD approach is modified in a plausible way (to handle missing data, among other issues), we obtain an algorithm that is equivalent to GROUSE.

II. GROUSE

The GROUSE algorithm was developed for identifying an unknown subspace $\mathcal{S}$ of dimension d in $\mathbb{R}^n$ from a sequence of vectors $v_t \in \mathcal{S}$ in which only the components indicated by the set $\Omega_t \subset \{1, \dots, n\}$ are revealed. Specifically, when $\bar{U}$ is an (unknown) $n \times d$ matrix whose orthonormal columns span $\mathcal{S}$, and $s_t \in \mathbb{R}^d$ is a weight vector, we observe the following subvector at iteration t:

$$(v_t)_{\Omega_t} = (\bar{U} s_t)_{\Omega_t} \tag{1}$$

(We use the subscript Ω_t on a matrix or vector to indicate restriction to the rows indicated by Ω_t.)

Algorithm 1 GROUSE

```
Given $U_0$, an $n \times d$ matrix with orthonormal columns, with $0 < d < n$;
Set $t := 1$;
repeat
    Take $\Omega_t$ and $(v_t)_{\Omega_t}$ from (1);
    Define $w_t := \arg\min_w \|[U_t]_{\Omega_t} w - [v_t]_{\Omega_t}\|_2^2$;
    Define $p_t := U_t w_t$; $[r_t]_{\Omega_t} := [v_t]_{\Omega_t} - [p_t]_{\Omega_t}$;
    $[r_t]_{\Omega_t^C} := 0$; $\sigma_t := \|r_t\| \, \|p_t\|$;
    Choose $\eta_t > 0$ and set
        $$U_{t+1} := U_t + (\cos(\sigma_t \eta_t) - 1) \frac{p_t}{\|p_t\|} \frac{w_t^T}{\|w_t\|} + \sin(\sigma_t \eta_t) \frac{r_t}{\|r_t\|} \frac{w_t^T}{\|w_t\|} \, . \qquad (2)$$
    $t := t + 1$;
until termination
```

GROUSE is described as Algorithm 1. It generates a sequence of $n \times d$ matrices U_t with orthonormal columns, updating with a rank-one matrix at each iteration in response to the newly revealed data $(v_t)_{\Omega_t}$. Note that GROUSE makes use of a steplength parameter η_t. It was shown in [2] that GROUSE exhibits local convergence of the range space of

978-1-4673-3146-3/13/$31.00 ©2013IEEE

U_t to the range space of $\bar{U}$, at an expected linear rate, under certain assumptions including incoherence of the subspace $\mathcal{S}$ with the coordinate directions, the number of components in Ω_t, and the choice of steplength parameter η_t.

III. Incremental Singular Value Decomposition

The incremental SVD algorithm of [4] computes the SVD of a matrix by adding one (fully observed) column at a time. The size of the matrices of left and right singular vectors U_t and V_t grows as columns are added, as does the diagonal matrix of singular values Σ_t. The approach is shown in Algorithm 2. Note that when the new vector v_t is already in the range space of U_t, we have $r_t = 0$, and the basic approach can be modified to avoid adding an extra dimension to the U, V, and Σ factors in this situation. If all vectors v_t lie in a subspace $\mathcal{S}$ of dimension d, the modified method will not need to grow U_t beyond size $n \times d$.

Algorithm 2 Incremental SVD [4]

```
Start with null matrixes $U_0$, $V_0$, $\Sigma_0$;
Set $t := 0$;
repeat
  Given new column vector $v_t$;
  Define $w_t := \arg\min_w \|U_t w - v_t\|_2^2 = U_t^T v_t$;
  Define
        $p_t := U_t w_t; \quad r_t := v_t - p_t;$
  (Set $r_0 := v_0$ when $t = 0$);
  Noting that
        $\begin{bmatrix} U_t\Sigma_t V_t^T & v_t \end{bmatrix} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \Sigma_t & w_t \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V_t & 0 \\ 0 & 1 \end{bmatrix}^T,$
  compute the SVD of the update matrix:
        $\begin{bmatrix} \Sigma_t & w_t \\ 0 & \|r_t\| \end{bmatrix} = \hat{U}\hat{\Sigma}\hat{V}^T, \qquad (3)$
  and set
        $U_{t+1} := \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \hat{U}, \quad \Sigma_{t+1} := \hat{\Sigma},$
        $V_{t+1} := \begin{bmatrix} V_t & 0 \\ 0 & 1 \end{bmatrix} \hat{V}.$
  $t := t + 1$;
until termination
```

IV. Relating GROUSE to Incremental SVD

Algorithms 1 and 2 are motivated in different ways and therefore differ in significant respects. We now describe a variant — Algorithm 3 — that is suited to the setting addressed by GROUSE, and show that it is in fact equivalent to GROUSE. Algorithm 3, includes the following modifications.

- Since only the subvector $(v_t)_{\Omega_t}$ is available, the missing components of v_t (corresponding to indices in the complement $\Omega_t^C := \{1, 2, \dots, n\} \setminus \Omega_t$) must be "imputed" from the revealed components and from the current subspace estimate U_t.
- The singular value matrix Σ_t is not carried over from one iteration to the next. In effect, the singular value estimates are all reset to 1 at each iteration.
- We allow an arbitrary rotation operator W_t to be applied to the columns of U_t at each iteration. This does not affect the range space of U_t, which is the current estimate of the underlying subspace $\mathcal{S}$.
- The matrix U_t is not permitted to grow beyond d columns.

Algorithm 3 ISVD for Partially Observed Vectors

```
Given $U_0$, an $n \times d$ matrix with orthonormal columns, with $0 < d < n$;
Set $t := 1$;
repeat
  Take $\Omega_t$ and $(v_t)_{\Omega_t}$ from (1);
  Define $w_t := \arg\min_w \|(U_t)_{\Omega_t} w - (v_t)_{\Omega_t}\|_2^2$;
  Define
        $[\tilde{v}_t]_i := \begin{cases} [v_t]_i & i \in \Omega_t \\ [U_t w_t]_i & i \in \Omega_t^C \end{cases};$
        $p_t := U_t w_t; \quad r_t := \tilde{v}_t - p_t;$
  Noting that
        $\begin{bmatrix} U_t & \tilde{v}_t \end{bmatrix} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} I & w_t \\ 0 & \|r_t\| \end{bmatrix},$
  we compute the SVD of the update matrix:
        $\begin{bmatrix} I & w_t \\ 0 & \|r_t\| \end{bmatrix} = \tilde{U}_t \tilde{\Sigma}_t \tilde{V}_t^T, \qquad (4)$
  and define $\hat{U}_t$ to be the $(d+1) \times d$ matrix obtained by removing the last column from $\tilde{U}_t$.
  Set $U_{t+1} := \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \hat{U}_t W_t$, where $W_t$ is an arbitrary $d \times d$ orthogonal matrix.
  $t := t + 1$;
until termination
```

Algorithm 3 is quite similar to an algorithm proposed in [3] (see Algorithm 4) but differs in its handling of the singular values. In [3], the singular values are carried over from one iteration to the next, but previous estimates are "down-weighted" to place more importance on the vectors $(v_t)_{\Omega_t}$ from recent iterations. This feature is useful in a scenario in which the underlying subspace $\mathcal{S}$ is changing in time. GROUSE also is influenced more by more recent vectors than older ones, thus has a similar (though less explicit) down-weighting feature.

We show now that for a particular choice of η_t in Algorithm 1, the Algorithms 1 and 3 are equivalent. Any difference in the updated estimate U_{t+1} is eliminated when we define the column rotation matrix W_t appropriately.

Theorem 1: Suppose that at iteration t of Algorithms 1 and 3, the quantities U_t, v_t, and Ω_t are the same, and that $w_t \neq 0$ and $r_t \neq 0$. Suppose the step size of GROUSE is

$$\eta_t = \frac{1}{\sigma_t} \arcsin \beta = \frac{1}{\sigma_t} \arccos(\alpha \|w_t\|), \qquad (5)$$

with α, β defined as follows:

$$\begin{aligned} \lambda :=& \frac{1}{2}(\|w_t\|^2 + \|r_t\|^2 + 1) + \\ & \frac{1}{2}\sqrt{(\|w_t\|^2 + \|r_t\|^2 + 1)^2 - 4\|r_t\|^2}, \end{aligned} \tag{6}$$

$$\beta := \frac{\|r_t\|^2 + \|w_t\|^2}{\|r_t\|^2 + \|w_t\|^2 + (\lambda - \|r_t\|^2)^2}, \tag{7}$$

$$\alpha := \frac{\|r_t\|(\lambda - \|r_t\|^2)}{\|r_t\|^2 + \|w_t\|^2 + (\lambda - \|r_t\|^2)^2}. \tag{8}$$

Define the $d \times d$ orthogonal matrix W_t by

$$W_t := \left[\frac{w_t}{\|w_t\|} \mid Z_t \right], \tag{9}$$

where Z_t is a $d \times d - 1$ matrix whose orthonormal columns span the orthogonal complement of w_t. For these choices of η_t and W_t, the iterates U_{t+1} generated by Algorithms 1 and 3 are identical.

Proof: We drop the subscript t freely throughout the proof. We first derive the structure of the matrix $\hat{U}_t$ in Algorithm 3, which is key to the update formula in this algorithm. We have from (4) that

$$\begin{bmatrix} I & w \\ 0 & \|r\| \end{bmatrix} \begin{bmatrix} I & 0 \\ w^T & \|r\| \end{bmatrix} = \begin{bmatrix} I + ww^T & \|r\|w \\ \|r\|w^T & \|r\|^2 \end{bmatrix} = \tilde{U}\tilde{\Sigma}^2\tilde{U}^T, \tag{10}$$

and thus the columns of $\tilde{U}$ are eigenvectors of this product matrix. We see that the orthonormal columns of the $d \times (d-1)$ matrix Z_t defined in (9) can be used to construct a set of eigenvectors that correspond to the eigenvalue 1, since

$$\begin{bmatrix} I + ww^T & \|r\|w \\ \|r\|w^T & \|r\|^2 \end{bmatrix} \begin{bmatrix} Z_t \\ 0 \end{bmatrix} = \begin{bmatrix} Z_t \\ 0 \end{bmatrix}. \tag{11}$$

Two eigenvectors and eigenvalues remain to be determined. Using λ to generally denote one of these two eigenvalues and $(y^T : \beta)^T$ to denote the corresponding eigenvector, we have

$$\begin{bmatrix} I + ww^T & \|r\|w \\ \|r\|w^T & \|r\|^2 \end{bmatrix} \begin{bmatrix} y \\ \beta \end{bmatrix} = \lambda \begin{bmatrix} y \\ \beta \end{bmatrix}. \tag{12}$$

The first block row of this expression yields $y + w(w^T y + \|r\|\beta) = \lambda y$, which implies that y has the form αw for some $\alpha \in \mathbb{R}$. By substituting this form into the two block rows from (12), we obtain

$$\begin{aligned} & \alpha(1 - \lambda)w + w(\alpha\|w\|^2 + \|r\|\beta) = 0 \\ \Rightarrow \quad & \alpha(1 + \|w\|^2 - \lambda) + \|r\|\beta = 0, \end{aligned} \tag{13}$$

and

$$\alpha\|r\|\|w\|^2 + (\|r\|^2 - \lambda)\beta = 0. \tag{14}$$

We require also that the vector

$$\begin{bmatrix} y \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha w \\ \beta \end{bmatrix}$$

has unit norm, yielding the additional condition

$$\alpha^2\|w\|^2 + \beta^2 = 1. \tag{15}$$

(This condition verifies the equality between the "$\arcsin$" and "$\arccos$" definitions in (5).)

To find the two possible values for λ, we seek non-unit roots of the characteristic polynomial for (10) and make use of the Schur form

$$\det\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = (\det D)\det(A - BD^{-1}C),$$

to obtain

$$\begin{aligned} \det & \begin{bmatrix} I + ww^T - \lambda I & \|r\|w \\ \|r\|w^T & \|r\|^2 - \lambda \end{bmatrix} \\ &= (\|r\|^2 - \lambda)\det\left[(1-\lambda)I + ww^T - \frac{\|r\|^2}{\|r\|^2 - \lambda}ww^T\right] \\ &= (\|r\|^2 - \lambda)\det\left[(1-\lambda)I - \frac{\lambda}{\|r\|^2 - \lambda}ww^T\right] \\ &= (1-\lambda)^d(\|r\|^2 - \lambda)\left(1 - \frac{\lambda\|w\|^2}{(\|r\|^2 - \lambda)(1 - \lambda)}\right) \\ &= (1-\lambda)^{d-1}\left((\|r\|^2 - \lambda)(1-\lambda) - \lambda\|w\|^2\right) \\ &= (1-\lambda)^{d-1}(\lambda^2 - \lambda(\|w\|^2 + \|r\|^2 + 1) + \|r\|^2), \end{aligned}$$

where we used $\det(I + aa^T) = 1 + \|a\|^2$. Thus the two non-unit eigenvalues are the roots of the quadratic

$$\lambda^2 - \lambda(\|w\|^2 + \|r\|^2 + 1) + \|r\|^2. \tag{16}$$

When $r \neq 0$ and $w \neq 0$, this quadratic takes on positive values at $\lambda = 0$ and when $\lambda \uparrow \infty$, while the value at $\lambda = 1$ is negative. Hence there are two roots, one in the interval $(0, 1)$ and one in $(1, \infty)$. We fix λ to the larger root, which is given explicitly by (6). The corresponding eigenvalue is the first column in the matrix $\tilde{U}_t$, and thus also in the matrix $\hat{U}_t$. It can be shown, by reference to formulas (6) and (16), that the values of β and α defined by (7) and (8), respectively, satisfy the conditions (13), (14), (15). We can now assemble the leading d eigenvectors of the matrix in (10) to form the matrix $\hat{U}$ as follows:

$$\hat{U} := \begin{bmatrix} \alpha w & Z_t \\ \beta & 0 \end{bmatrix}.$$

Thus, with W_t defined as in (9), we obtain

$$\hat{U}W_t^T = \begin{bmatrix} \alpha w & Z_t \\ \beta & 0 \end{bmatrix} \begin{bmatrix} \frac{w^T}{\|w\|} \\ Z_t^T \end{bmatrix} = \begin{bmatrix} \frac{\alpha}{\|w\|}ww^T + Z_t Z_t^T \\ \frac{\beta}{\|w\|}w^T \end{bmatrix}.$$

Therefore, we have from the update formula for Alg 3 that

$$\begin{aligned} U_{t+1} &= \begin{bmatrix} U_t & \frac{r}{\|r\|} \end{bmatrix} \hat{U}W_t^T \\ &= U_t\left(\frac{\alpha}{\|w\|}ww^T + Z_t Z_t^T\right) + \beta\frac{r}{\|r\|}\frac{w^T}{\|w\|}. \end{aligned}$$

By orthogonality of W_t, we have

$$I = WW^T = \frac{ww^T}{\|w\|^2} + Z_t Z_t^T \ \Rightarrow \ Z_t Z_t^T = I - \frac{ww^T}{\|w\|^2}.$$

Hence, by substituting in the expression above, we obtain

$$\begin{aligned} U_{t+1} &= U_t\left(\alpha\frac{ww^T}{\|w\|} + \left(I - \frac{ww^T}{\|w\|^2}\right)\right) + \beta\frac{r}{\|r\|}\frac{w^T}{\|w\|} \\ &= U_t + \left[(\alpha\|w\| - 1)\frac{w}{\|w\|} + \beta\frac{r}{\|r\|}\right]\frac{w^T}{\|w\|}, \end{aligned}$$

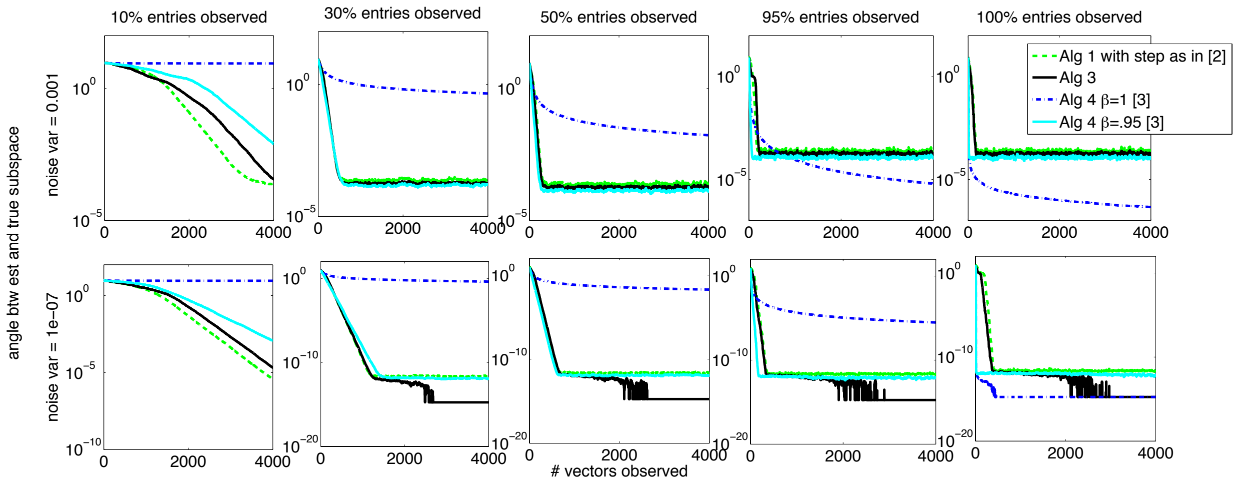

Fig. 1. Results for the algorithms described in this paper. Algorithm 4 with $\beta = 1$ and full data is equivalent to the original incremental SVD (Algorithm 2). This algorithm performs the best when all entries are observed or when just a small amount of data is missing and noise is present. Algorithm 4 with $\beta = 0.95$ and full data at first converges quickly as with $\beta = 1$ but flatlines much earlier. GROUSE (Algorithm 1) with the step as prescribed in [2] does the best when a very small fraction of entries are observed, approaching the theoretical minimum (see [2] for details). With low noise and missing data, our ISVD method (Algorithm 3) averages out the noise, given enough iterations. Otherwise the algorithms perform equivalently.

which is identical to the update formula in Algorithm 1 provided that

$$\cos \sigma_t \eta_t = \alpha \|w_t\|, \quad \sin \sigma_t \eta_t = \beta.$$

These relationships hold because of the definition (5) and the normality relationship (15). ■

Algorithm 4 Another ISVD approach for Partial Data [3]

```
Given $U_0$, an arbitrary $n \times d$ matrix with orthonormal columns, with $0 < d < n$; $\Sigma_0$, a $d \times d$ diagonal matrix of zeros which will later hold the singular values.
Set $t := 1$;
repeat
    Compute $w_t, p_t, r_t$ as in Algorithm 3.
    Compute the SVD of the update matrix:
        $\begin{bmatrix} \beta\Sigma_t & w_t \\ 0 & \|r_t\| \end{bmatrix} = \hat{U}\hat{\Sigma}\hat{V}^T,$
    for some scalar $\beta \leq 1$ and set
        $U_{t+1} := \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \hat{U}, \quad \Sigma_{t+1} := \hat{\Sigma}.$
    $t := t + 1$;
until termination
```

V. Simulations

To compare the algorithms presented in this note, we ran simulations as follows. We set $n = 200$ and $d = 10$, and defined $\bar{U}$ (whose columns span the target subspace $\mathcal{S}$) to be a random matrix with orthonormal columns. The vectors v_t were generated as $\bar{U}s_t$, where the components of s_t are $\mathcal{N}(0, 1)$ i.i.d. We also computed a different $n \times d$ matrix with orthonormal columns, and used that to initialize all algorithms. We compared the GROUSE algorithm (Algorithm 1) with our proposed missing data ISVD (Algorithm 3). Although, as we show in this note, these algorithms are equivalent for a particular choice of η_t, we used the different choice of this parameter prescribed in [2]. Finally, we compared to the incomplete data ISVD proposed in [3], which is summarized in Algorithm 4. This approach requires a parameter β which down-weights old singular value estimates. We obtained the performance for $\beta = 0.95$; performance of this approach degraded for values of β less than 0.9. The error metric on the y-axis is $d - \|U_t^T \bar{U}\|_F^2$; see [2] for details of this quantity.

VI. Conclusion

We have shown an equivalence between GROUSE and a modified incremental SVD approach. The equivalence is of interest because the two methods are motivated and constructed from different perspectives — GROUSE from an optimization perspective, and incremental SVD from linear algebra perspective.

References

[1] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proceedings of the Allerton conference on Communication, Control, and Computing*, 2010.

[2] Laura Balzano and Stephen J. Wright. Local convergence of an algorithm for subspace identification from partial data. Submitted for publication. Preprint available at http://arxiv.org/abs/1306.3391.

[3] M. Brand. Incremental singular value decomposition of uncertain data with missing values. *European Conference on Computer Vision (ECCV)*, pages 707–720, 2002.

[4] James R. Bunch and Christopher P. Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 31:111–129, 1978. 10.1007/BF01397471.

[5] Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[6] Pierre Comon and Gene Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8), August 1990.

[7] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.