

# CLUSTERING OF NONNEGATIVE DATA AND AN APPLICATION TO MATRIX COMPLETION

*C. Strohmeier, D. Needell*

Department of Mathematics, University of California, Los Angeles

## ABSTRACT

In this article, we propose a simple algorithm to cluster nonnegative data lying in disjoint subspaces. We analyze its performance in relation to a certain measure of correlation between said subspaces. We use our clustering algorithm to develop a matrix completion algorithm which can outperform standard matrix completion algorithms on data matrices satisfying a certain natural low rank condition.

**Index Terms**— Nonnegative matrix factorization, matrix completion, clustering, compressed classification

## I. INTRODUCTION

Real world data is often high dimensional, that is to say a given data point may be modeled by a vector in a high dimensional Euclidean space. While genuinely high dimensional data would be prohibitively difficult to analyze, most data encountered are effectively low dimensional. For instance, the data points may approximately lie in some (a priori unknown) low dimensional subspace of their ambient space. Moreover, one often encounters data which is non-negative in the sense that each of its entries is nonnegative, such as data stemming from user surveys, rating systems, or biomedical monitoring. Here, we consider such low dimensional data and address the problem of data completion via two other data oriented tasks, clustering and nonnegative matrix factorization.

An important problem in data science is that of data completion, and matrix completion in particular [1]. Namely, one wishes to recover an unknown matrix from only a subset of its entries. To make the problem well-posed, one typically assumes that the underlying matrix is also low-rank, a reasonable assumption in many applications where there is a small number of intrinsic features that describe the large-scale data. One typically also assumes that the observed entries are selected in a “nice” way, such as uniformly at random, to avoid degenerate sampling patterns. There are now many provably robust methods to matrix completion including convex optimization programs [2], [3] and projection based methods [4], [5].

*Clustering* is another typical problem in data science whose aim is to cluster, or group, unlabeled data. That is,

Partially supported by NSF CAREER #1348721, and NSF BIGDATA #1740325.

one has a data set consisting of two or more families of data points such that members of each family share intrinsic characteristics. Based on these intrinsic characteristics, one must sort the data into its different families. There are now many methods to cluster data along with a wide array of theoretical and empirical support, see e.g. [6], [7], [8] and references therein. Although there are now many sophisticated methods for clustering, the simpler  $k$ -means clustering method [9], which aims to separate the data points into  $k$  clusters so that each point belongs to a cluster with the nearest mean, is still useful in many applications. However, like many others,  $k$ -means fails in most applications where the data families live in some low dimensional subspaces, where linear separability need not be apparent.

Lastly, another useful tool we will utilize is nonnegative matrix factorization (NMF) [10], [11]. Concretely, the problem of NMF is to factor an  $M \times N$  data matrix  $X$  into  $X \approx AS$  where  $A$  is a non-negative  $M \times T$  matrix and  $S$  a non-negative  $T \times N$  matrix. The parameter  $T$  corresponds to the number of *topics* to represent the data, the matrix  $A$  then gives a topic representation for each of the  $M$  variables, and  $S$  a topic representation for each of the  $N$  users (for example). Concretely, we can view  $A_{ij}$  as an indicator of how important the  $i$ th variable is for the  $j$ th topic, and  $S_{jk}$  as how important the  $j$ th topic is to user  $k$ . This structure implicitly reveals topics in the data, which can be interpreted on their own or used as features in other data processing tasks. This type of NMF is called *unsupervised* representation since it works only on the raw data, without any other observation information. NMF is also by now a standard tool in dimensionality reduction. The advantage of this factorization compared to other dimensionality reduction techniques such as PCA, IMF, etc. is that the nonnegativity constraints enforce a certain locality, and hence interpretability, of its hidden features. See for instance [12], [13].

In this paper, we discuss the clustering and completion problems in the context of nonnegative data belonging to disjoint low dimensional subspaces of a high dimensional Euclidean space. This additional structure appears in data arising from an abundance of applications, ranging from collaborative filtering to multi-class learning, where the structure may arise from subgroups of e.g. users that have similar preferences. Utilizing this additional structure allows

us to decrease the number of observations needed and/or decrease the reconstruction error. In particular, we present a clustering algorithm based on nonnegative matrix factorization (NMF), Algorithm 1, and examine the relationship between its performance and a certain measure of correlation between the two families of data points that we are trying to separate (cluster). We then discuss an application to a certain "block completion" algorithm, which can yield significant improvements compared to standard matrix completion algorithms regarding recovery error for matrices satisfying a very natural low-rank type condition.

## II. CLUSTERING VIA NMF

Consider a data set consisting of vectors  $x_i$  belonging to one of two disjoint low dimensional subspaces  $W_1, W_2 \leq \mathbb{R}^n$ . Suppose further that the entries of the data vectors are all nonnegative. The problem is to sort the data according to its respective subspace. While standard clustering techniques applied to the data fail in general, one may take advantage of a certain orthogonalizing effect of NMF. To explain the first step in our proposed simple method, let us interpret our  $m$  data points  $x_i \in \mathbb{R}^n$  as row vectors and concatenate them into the data matrix  $X \in \mathbb{R}^{m \times n}$ . Suppose we have an upper bound for the sum of the dimensions of the subspaces:

$$\dim W_1 + \dim W_2 \leq r$$

We perform NMF with  $r$  hidden features (topics) to factor  $X$  into a product of a weight matrix  $W \in \mathbb{R}^{m \times r}$  and a hidden feature matrix  $H \in \mathbb{R}^{r \times n}$ . We then cluster not on the original data, but on the rows of the weight matrix  $W$ . This is described succinctly as Algorithm 1.

---

### Algorithm 1 Clustering Via NMF

---

1. Input: Data matrix  $X$ , upper bound for rank  $r$ .
  2. NMF: Perform NMF with  $r$  hidden features to write  $X = WH$
  3. Apply k-means to rows of  $W$
- 

Algorithm 1 proves to be very effective, even in the presence of considerable noise. One explanation for this phenomenon is that NMF exhibits a certain orthogonalization effect. In particular, it is not necessarily true that if one forms a nonnegative matrix  $H'$  whose rows are a given set of  $r$  linearly independent nonnegative vectors belonging to  $W_1 + W_2$  that one may still be able to factor  $X = W'H'$  for some *nonnegative*  $W'$ . This prevents a degree of "mixing" of hidden feature basis vectors associated to the two subspaces when performing NMF. The result is that the rows of any weight matrix  $W$  obtained via NMF applied to our data matrix will have very small values at entries corresponding to basis vectors for a family in which it does not belong. Put simply, NMF has reduced the problem of clustering nonnegative data belonging to general low dimensional subspaces to that of clustering nonnegative data

belonging to low dimensional *coordinate* subspaces (thus NMF has "orthogonalized" the original subspaces).

Below we illustrate the relationship between success of clustering and a certain measure of correlation between subspaces.

Suppose  $U, V \subset \mathbb{R}^n$  are two subspaces of the same dimension  $r$ . We define the correlation measure  $\alpha(U, V)$  between them as

$$\alpha(U, V) = \frac{1}{r} \text{tr}(P_U P_V),$$

where  $P_U$  denotes the orthogonal projection onto  $U$  and similarly for  $P_V$ . Note that for any two subspaces we have  $0 \leq \alpha(U, V) \leq 1$ ,  $\alpha(U, V) = 0$  iff  $U$  and  $V$  are orthogonal and  $\alpha(U, V) = 1$  iff  $U = V$ .

Naturally, one expects that as the correlation measure between subspaces increases, they become more difficult to cluster. This is indeed the case.

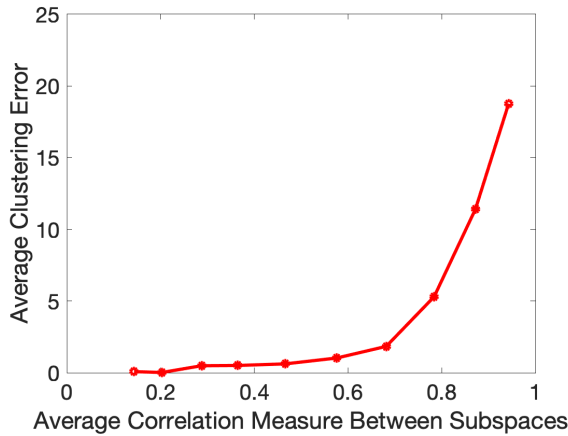
Fix  $n, r$  with  $r \ll n$  and consider an  $r \times n$  random matrix with i.i.d. entries uniformly distributed between 0 and 1. A particular instance of this random matrix is a proxy for an  $r$ -dimensional subspace of  $\mathbb{R}^n$ . Abusing notation, we will refer to such an instance as  $U$ . We construct matrices  $V$  with various correlation measures through a multiplication by a matrix of the form  $\exp(tA)$  where  $t$  is a scalar and  $A$  is a random skew-symmetric matrix (obtained by skew-symmetrizing an instance of a random standard Gaussian matrix). Thus  $t = 0$  implies  $U = V$ , and larger  $t$  implies a greater correlation measure between the subspaces. Of course  $t$  cannot be too large, otherwise  $V$  may possess negative entries.

We will consider various pairs of  $U$  and  $V$  as constructed above. Let  $U$ -block be the product of an instance of an  $m \times r$  standard uniform matrix with  $U$ , and construct  $V$ -block similarly. We concatenate these two row matrices to obtain our  $2m \times n$  row matrix  $X$ .

Figure 1 represents the average clustering error under the above model with  $m = 100, n = 80, r = 5$ . Each simulation consists of 100 samples. As noted, the accuracy is smaller for smaller values of  $\alpha$ , when the subspaces are closer to being orthogonal.

## III. APPLICATION TO MATRIX COMPLETION

In many applications where one wishes to perform data completion, the data points naturally lie in different subspaces. For example, in collaborative filtering where the data represents users and their ratings of certain products, the overall data matrix may be low rank since there are a few underlying features that describe the users or products (e.g. movie genre, user demographics or preferences, etc.). However, it may more often be the case that even more structure is present, for example the users may be divided into similar blocks so that each block is of even lower rank than the overall matrix. If such blocks can be identified, performing



**Fig. 1:** Average clustering error (average number of misclassified data points over 100 trials) as a function of the average correlation measure  $\alpha(U, V) = \frac{1}{r} \text{tr}(P_U P_V)$  between the subspaces  $U, V$  which generated the data.

matrix completion on each block individually will result in lower error and/or the need for fewer measurements.

Here we thus consider the problem of matrix completion, applied to matrices of the form considered in the previous section (namely concatenations of blocks). In fact we allow for matrices obtained by permuting the rows of such matrices, since the data is unlabeled. While one can attempt to apply standard matrix completion, we demonstrate that improvements can be made if one exploits the additional structure of our data matrices.

We introduce a novel matrix completion algorithm, Algorithm 2, which we refer to as Block Completion. The algorithm proceeds by first applying an initial, "basic" matrix completion to the entire data matrix.

In practice, applying a finite number of iterates of a standard matrix completion algorithm may yield matrices which have negative entries, even for nonnegative, low rank matrices, let alone for matrices which are only approximately low rank. We avoid this problem by using a certain simultaneous matrix completion and nonnegative matrix factorization algorithm, MC-NMF, as in [14], to implement our basic completion. MC-NMF was formulated to minimize a non-convex, constrained least squares objective, and is based on the standard alternating direction method of multipliers (ADMM) framework. The algorithm takes in an incomplete, low rank, nonnegative matrix  $X$ , a set of sampled values, and an estimate for its rank, and at each iterate returns a completed pair of low rank nonnegative factors  $W, H$  such that after sufficiently many iterates one obtains  $X \approx WH$ . In short, the "basic completion" algorithm used in this paper simply applies MC-NMF with a fixed number of iterates and returns the product  $WH$  of its outputs.

Although the error from this initial matrix completion

may be large, empirically one often finds that Algorithm 1 nonetheless succeeds in clustering the (noisy) data points. We obtain data sub-matrices (blocks) from these clusters, and apply basic completion to each block using the original observed entries. The point is that the blocks are lower rank, and so one expects an improvement in recovery error. We concatenate the completed blocks in the obvious way and take this as the output for block completion.

---

#### Algorithm 2 Block Completion

---

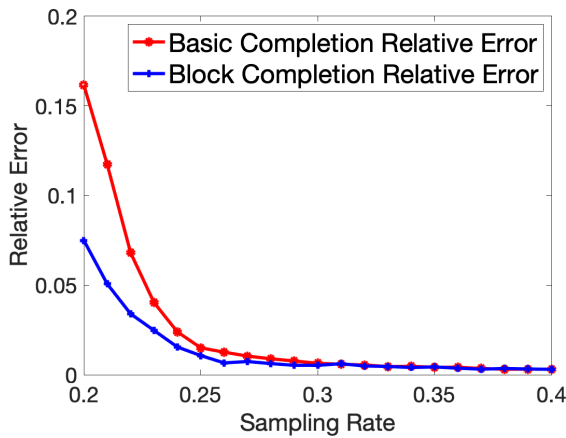
1. Initialization: Mask  $\Omega$  of observed entries of data matrix  $X$ , upper bound for rank  $r$ .
  2. Basic Completion: Apply standard matrix completion algorithm to whole matrix  $X$  using observed entries  $\Omega$ .
  3. Sort Blocks: Apply clustering via NMF (Algorithm 1) to obtain matrices  $A$  and  $B$  by concatenating the data vectors belonging to respective clusters.
  4. Complete Blocks: Using masks derived from original mask, apply basic completion to both  $A$  and  $B$  individually.
  5. Reassemble: construct the full completed matrix from the completed blocks.
- 

To test this approach, we recorded a comparison of standard completion with block completion. Our data matrices were constructed by concatenating two low rank blocks. Each block was the product of an instance of a standard uniform  $100 \times 5$  matrix with a standard uniform  $5 \times 80$  matrix. The observed entries corresponded to a Bernoulli matrix with various sampling rates  $p$ . We used 500 iterates of the basic completion algorithm described above. The result represents the average relative error of one hundred random samples. Figure 2 demonstrates the advantages of such an approach. Indeed, for low sampling rates the relative error when applying block completion is significantly lower than that of basic completion. Figure 4 shows an example of the obtained results.

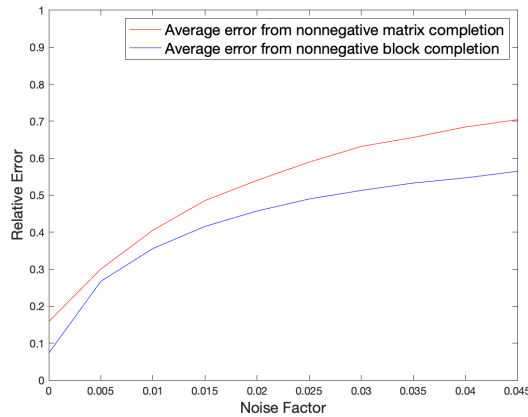
We empirically confirm that our method is robust to a reasonable level of noise. In 3, we fix a sampling rate of .2 and consider various noise levels. Our data has the form of the data matrices described above, plus a noise term. The noise is simulated as a standard uniform matrix, normalized so that its frobenius norm is the product of the noise level factor indicated on the x-axis of the figure by the frobenius norm of the exact matrix. The indicated values associated to each noise level factor are the results of averaging the relative errors of 100 trials each.

## IV. CONCLUSION

We have presented a simple clustering algorithm based on NMF which is very effective in separating nonnegative data belonging to disjoint, low-dimensional subspaces, even with considerable noise. As an application, we introduced a matrix completion algorithm designed to more accurately



**Fig. 2:** Comparison of Relative Error Between Completion Algorithms as a function of the sampling rate  $p$ .

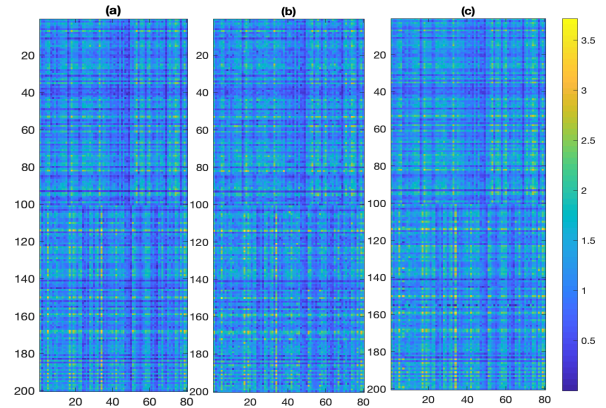


**Fig. 3:** Comparison of Relative Error Between Completion Algorithms as a function of noise factor.

complete certain matrices satisfying a low-rank type condition. This algorithm has the potential to yield non-trivial insights from incomplete data matrices comprised of data points belonging to two or more different families.

## V. REFERENCES

- [1] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.
- [2] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proc. 22nd int. conf. on Machine learning*. 2005, pp. 713–719, ACM.
- [3] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 9, no. 6, pp. 925–936, 2010.
- [4] R. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.
- [5] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh, "Matrix completion from noisy entries," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2057–2078, 2010.
- [6] Anil K Jain, M Narasimha Murty, and Patrick J Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
- [7] Rui Xu and Donald C Wunsch, "Survey of clustering algorithms," 2005.
- [8] Salem Alelyani, Jiliang Tang, and Huan Liu, "Feature selection for clustering: A review," in *Data Clustering*, pp. 29–60. Chapman and Hall/CRC, 2018.
- [9] M David, "Chapter 20. an example inference task: clustering," *Information Theory, Inference and Learning Algorithm*, 2003.
- [10] Rashish Tandon and Suvrit Sra, "Sparse nonnegative matrix approximation: new formulations and algorithms," 2010.
- [11] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational statistics & data analysis*, vol. 52, no. 1, pp. 155–173, 2007.
- [12] Daniel D Lee and H Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.
- [13] Daniel D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788, 1999.
- [14] Yangyang Xu, Wotao Yin, Zaiwen Wen, and Yin



**Fig. 4:** Example of original matrix (a), reconstructed matrix via standard matrix completion (b), and by the proposed block completion method (c). Errors for standard and proposed approach are .17 and .05, respectively.

Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Frontiers of Mathematics in China*, vol. 7, no. 2, pp. 365–384, 2012.